LOAD DISTRIBUTION METHOD FOR CLUSTER TYPE COMPUTER DEVICE

Patent number: JP9160885
Publication date: 1997-06-20
Inventor: MAYA YUZURU; UGAJIN ATSUSHI; KINOSHITA TOSHIYUKI
Applicant: HITACHI LTD
Classification:
- international: G06F15/16
- european:
Application number: JP19950316339 19951205
Priority number(s): JP19950316339 19951205

View INPADOC patent family

Abstract of **JP9160885**

PROBLEM TO BE SOLVED: To level the loads on processor modules and improve the throughput of the whole system by making a communication processor module calculate the process times of a transaction processor module and a file processor module and send a message to a reception-side processor module. SOLUTION: The function for the message process is divided into a communication process for a communication with a terminal, a transaction process, and a file process for controlling access to a disk. Transaction and file processor modules 14-16, and 17-19 send system operation information to communication processor modules 11-13 through a communication path 1 for control. The communication processor modules 11-13, on the other hand, calculate the process times of the reception modules and selects the processor modules having the shortest process time, thereby sending the message to the reception-side processor module.

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平9－160885

(43)公開日　平成９年(1997)６月20日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ		技術表示箇所
Ｇ０６Ｆ　15/16	３８０		Ｇ０６Ｆ　15/16	３８０Ｚ	

審査請求　未請求　請求項の数20　ＯＬ　（全 31 頁）

(21)出願番号　特願平7－316339

(22)出願日　平成７年(1995)12月５日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　真矢　譲
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72)発明者　宇賀神　敦
神奈川県海老名下今泉810番地　株式会社日立製作所オフィスシステム事業部内

(72)発明者　木下　俊之
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(74)代理人　弁理士　小川　勝男

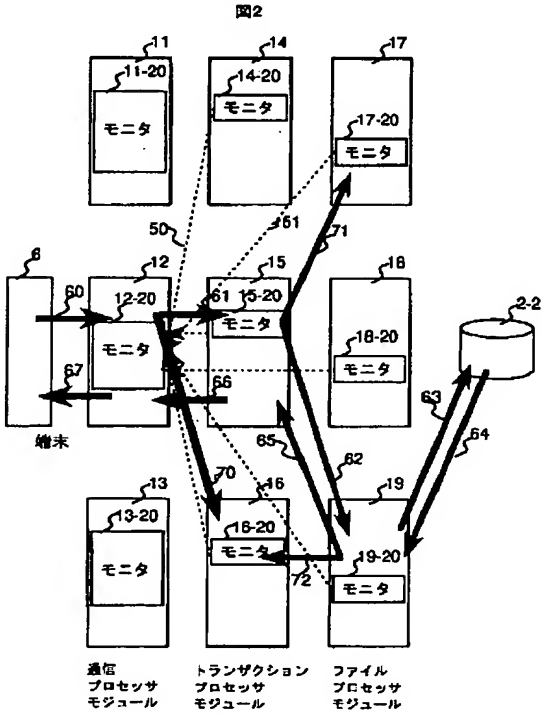
(54)【発明の名称】　クラスタ型計算機装置の負荷分散方法

(57)【要約】

【課題】通信プロセッサモジュールは処理時間を算出し、受信側プロセッサモジュールに電文を送信することにより、処理能力を向上させ、障害時の回復時間を短縮する。

【解決手段】プロセッサモジュールは、プロセッサ、メモリおよびIOPから構成し、プロセス間通信パスと制御用通信パスにより接続する。電文処理は、端末との通信を行う通信プロセス、トランザクションプロセスおよびファイルプロセスに分割する。トランザクションとファイルプロセッサモジュール（通信プロセッサモジュール）は、制御用通信パスを介して、通信プロセッサモジュールにシステム稼働情報を送信する。一方、通信プロセッサモジュールは、電文を送信する際、受信側プロセッサモジュールの処理時間を計算し、処理時間より受信側プロセッサモジュールとその予備のプロセッサモジュールを選び、電文を送信する。

図2



【特許請求の範囲】

【請求項１】それぞれがプロセッサ、メモリ、及びIOを制御するIOPからなる複数のプロセッサモジュールをプロセス間通信パスおよび制御用通信パスにより接続したシステムにおいて、

電文処理は、通信プロセス、トランザクションプロセスおよびファイルプロセスに分割してそれぞれを各プロセッサモジュールに搭載することによって通信プロセッサモジュール、トランザクションプロセッサモジュール、及びファイルプロセッサモジュールを構成し、前記トランザクションプロセッサモジュールおよび前記ファイルプロセッサモジュールは前記制御用通信パスを介して前記通信プロセッサモジュールにシステム稼働情報を送信し、通信プロセッサモジュールがシステム稼働情報を管理することを特徴とするクラスタ型計算機装置の負荷分散方法。

【請求項２】上記システムにおいて、前記トランザクションプロセッサモジュールと前記ファイルプロセッサモジュールは、システム立ち上げ時、あるいはプロセッサモジュール追加時に、自身のCPU処理能力、IO待ち時間およびプロセス間通信の実行ステップ数を前記通信プロセッサモジュールに送信することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項３】上記システムにおいて、前記プロセッサモジュールは多数の電文をいくつかのタイプに分け、電文のタイプ毎にシステム稼働情報として使用することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項４】上記システムにおいて、前記トランザクションプロセッサモジュールと前記ファイルプロセッサモジュールは、一定間隔毎に、システム稼働情報として、待ち行列に格納されている電文のタイプとその個数を、前記通信プロセッサモジュールに送信することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項５】上記システムにおいて、前記通信プロセッサモジュールは、受信した待ち行列に格納されている電文のタイプとその個数、CPU処理能力、IO待ち時間、プロセス間通信の実行ステップ数から、トランザクションプロセッサモジュールとファイルプロセッサモジュールからなる受信側プロセッサモジュールのすべてについて処理時間を算出し、処理時間が最小となるトランザクションプロセッサモジュールとファイルプロセッサモジュールを選び、前記通信プロセッサモジュールは、トランザクションプロセッサモジュールに電文と送信すべき前記ファイルプロセッサモジュールのアドレスを送信することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項６】上記システムにおいて、前記トランザクションプロセッサモジュールは、前記通信プロセッサモジュールからの電文と前記ファイルプロセッサモジュールのアドレスを受信し、前記アドレスを参照して、前記ファイルプロセッサモジュールに電文を送信することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項７】上記システムにおいて、前記通信プロセッサモジュールはすべての前記トランザクションプロセッサモジュールと前記ファイルプロセッサモジュールの処理時間を算出し、前記処理時間の値が一定値以上ならば、輻輳状態と判定し、前記通信プロセッサモジュールは輻輳状態のプロセッサモジュールへの電文送信を禁止することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項８】上記システムにおいて、前記通信プロセッサモジュールが算出した処理時間が一定値以上のトランザクションプロセッサモジュールあるいはファイルプロセッサモジュールが輻輳状態に遷移した場合に、前記トランザクションプロセッサモジュールあるいは前記ファイルプロセッサモジュールは、前記通信プロセッサモジュールに対して電文の送信禁止を通知することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項９】上記システムにおいて、前記通信プロセッサモジュールはすべての前記トランザクションプロセッサモジュールと前記ファイルプロセッサモジュールの処理時間を算出し、前記処理時間の値が一定値以下ならば、輻輳状態から解除したと判定し、前記通信プロセッサモジュールは輻輳状態から解除した前記トランザクションプロセッサモジュールあるいは前記ファイルプロセッサモジュールへの電文送信を再開することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項１０】上記システムにおいて、前記通信プロセッサモジュールは、輻輳状態の前記トランザクションプロセッサモジュールあるいは前記ファイルプロセッサモジュールの処理時間が一定値以下になった場合に、前記トランザクションプロセッサモジュールあるいは前記ファイルプロセッサモジュールは、前記通信プロセッサモジュールに対して電文の送信禁止を解除することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項１１】上記システムにおいて、前記プロセッサモジュールが輻輳状態に遷移、あるいは輻輳状態から解除されたことを、前記制御用通信パスに接続されたコンソールに表示することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。

【請求項１２】上記システムにおいて、前記トランザクションプロセッサモジュールあるいは前記ファイルプロセッサモジュールで障害あるいは輻輳が発生した場合に、前記通信プロセッサモジュールは再度処理時間を計

算し、電文を再送することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。
【請求項１３】上記システムにおいて、前記トランザクションプロセッサモジュールあるいは前記ファイルプロセッサモジュールで障害あるいは輻輳状態から回復すると、前記通信プロセッサモジュールは再度処理時間を計算し、電文を送信することを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。
【請求項１４】上記システムにおいて、前記通信プロセッサモジュールは、処理時間が最小であって、電文を送信すべきトランザクションプロセッサモジュールとファイルプロセッサモジュールの次に処理時間の小さいトランザクションプロセッサモジュールとファイルプロセッサモジュールを予備のプロセッサモジュールとして稼働させて電文を送信し、障害が発生すると、前記予備のプロセッサモジュールが処理を引き継ぐことを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。
【請求項１５】上記システムにおいて、前記トランザクションプロセッサモジュールあるいはファイルプロセッサモジュールで障害が発生した場合に、前記予備のトランザクションプロセッサモジュールあるいはファイルプロセッサモジュールは、自身が受信している電文を参照することにより、前記通信プロセッサモジュールからの電文の再送を要求しないで引き継ぎ処理を実行することを特徴とする請求項１４記載のクラスタ型計算機装置の負荷分散方法。
【請求項１６】上記システムにおいて、障害から回復すると、前記トランザクションプロセッサモジュールあるいは前記ファイルプロセッサモジュールは前記通信プロセッサモジュールに障害回復を通知し、処理時間を算出することを特徴とする請求項１４記載のクラスタ型計算機装置の負荷分散方法。
【請求項１７】上記システムにおいて、前記プロセッサモジュールのそれぞれは複数のプロセスを搭載し、１つのプロセスを現用プロセスとして稼働させ、他のプロセスを予備プロセスとして待機させ、あるプロセッサモジュールで障害が発生した場合に、障害プロセッサモジュールを閉塞し、さらに更にシステム処理能力が最高になるように、前記障害プロセスを他の機能のプロセッサモジュールが引き継ぐことを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。
【請求項１８】上記システムにおいて、電文をn個のプロセスに分割したn階層のクラスタ型計算機を構成し、最上位プロセッサモジュールが下位のプロセッサモジュールの処理時間を算出し、これが最小となるプロセッサモジュールで電文処理を実行させることを特徴とする請求項１記載のクラスタ型計算機装置の負荷分散方法。
【請求項１９】上記システムにおいて、前記最上位プロセッサモジュールが下位のプロセッサモジュールの処理時間を算出し、前記処理時間が２番目に小さいプロセッサモジュールを予備のプロセッサモジュールとして稼働させ、前記予備のプロセッサモジュールにも電文を送信し、障害時および輻輳時に前記予備のプロセッサモジュールが処理を引き継ぐことを特徴とする請求項１６記載のクラスタ型計算機装置の負荷分散方法。
【請求項２０】上記システムにおいて、前記最上位プロセッサモジュールが前記通信プロセッサモジュールの機能を実行することを特徴とする請求項１７記載のクラスタ型計算機装置の負荷分散方法。
【発明の詳細な説明】
【０００１】
【発明に属する技術分野】本発明は、複数のプロセッサモジュールをプロセス間通信パスおよび制御用通信パスにより接続し、電文処理を複数のプロセスに分割し、分割されたプロセスを複数のプロセッサモジュールに搭載したクラスタ型計算機システムに係わり、プロセッサモジュールの負荷を均一にすることにより、システム全体の処理能力を向上させ、更に障害時におけるシステム回復時間を短縮するクラスタ型計算機装置およびその負荷分散方法に関する。
【０００２】
【従来の技術】計算機システムはダウンサイジングにより、汎用大型機を中心としたシステムから多数のプロセッサモジュールを高速バスにより接続する分散システムに移行している。一方、計算機システムの処理形態はオンライントランザクション処理が主流になり、年々トランザクション処理量が増大し、高い処理能力が要求されている。
【０００３】ところで、従来の分散機では、日経エレクトロニクス（1992.5.18、No.554、p.87～p.96）に記載されている。これは、現用系と待機系からなるホットスタンバイ構成であり、これらの系間に制御用通信パスを設け、障害を検出するためのaliveメッセージのみを送信していた。しかし、このようなホットスタンバイシステムでは電文の処理時間やIO（入出力）待ち時間のようなシステム稼働情報を他系に送信していなかった。
【０００４】一方、従来の負荷分散方法は、”マルチプロセッサ時分割システムにおける負荷分散アルゴリズム”（電子情報通信学会論文誌 D-I Vol.J72-D-I No.2 PP.75-82 1989年2月）に記載されているように、２つのプロセッサモジュール間でCPU利用率あるいは応答時間のようなシステム稼働情報により、プロセッサモジュール間の負荷分散を行っていた。
【０００５】最後に、並列計算機でも同様の技術が行われており、「並列計算機構成論」、冨田真治著（p.195～p.200）では、以下のように記載されている。すなわち、並列計算機では、フェールセーフシステムの適用により、1プロセッサに複数の機能を持たせない。プロセッサ障害時には、処理を引き継ぐのではなく、障害プロセッサを閉塞させている。

【０００６】SMS-201の場合、動作モードは制御、通信および実行の各モードに分割されている。制御・通信モードでは、マスタプロセッサ（メインプロセッサ）は各プロセッサの結果が格納されている通信用メモリの出力バッファの内容を読み出し、それをすべてのプロセッサの通信用メモリの入力バッファにブロードキャスト方式により書き込む。従って、各プロセッサの通信用メモリの内容はすべて同一になる。実行モードでは、各プロセッサは通信メモリを使用して、相互に全く独立に動作し、結果を通信用メモリの出力バッファに格納し、処理の終了を主プロセッサに報告し、制御・通信モードを待つ。

【０００７】

【発明が解決しようとする課題】クラスタ型計算機システムにおいて、オンライントランザクション処理を実行する場合には、複数のプロセッサモジュールをプロセス間通信パスおよび制御用通信パスにより接続し、電文処理を複数のプロセス（通信プロセス、トランザクションプロセスおよびファイルプロセス）に分割する。そして、分割されたプロセスはそれぞれ多数のプロセッサモジュールで実行される。

【０００８】そして、通信プロセッサモジュールが端末から電文を受信すると、トランザクションプロセッサモジュールに電文を送信する。そして、トランザクションプロセッサモジュールは電文を解析し、ファイルプロセッサモジュールにディスクの読み出し処理や書き込み処理を要求する。更に、ファイルプロセッサモジュールはディスクへの読み出し処理や書き込み処理を実行する。その後、ファイルプロセッサモジュールは、これらの処理が終了すると、ディスクのアクセス要求の送信元のトランザクションプロセッサモジュールにディスクへの書き込み完了を通知する。そして、トランザクションプロセッサモジュールは、電文送信元の通信プロセッサモジュールに処理結果を送信する。最後に、通信プロセッサモジュールは処理結果を端末に通知する。

【０００９】このように、トランザクション処理では、ディスクの参照および更新に関し、通信プロセッサモジュールからファイルプロセッサモジュールまでの処理の経路では、行きと帰りの経路は同一のプロセッサモジュールでなければならない。

【００１０】ところで、従来の分散機構では、現用系と待機系からなるホットスタンバイ構成であり、これらの系間に制御用通信パスを設け、障害を検出するためのaliveメッセージのみを送信していた。しかし、このようなホットスタンバイシステムでは電文の処理時間やIO待ち時間のようなシステム稼働情報を他系に送信していなかった。その結果、負荷の分散は考慮されていなかった。

【００１１】次に、従来の負荷分散方法は、”マルチプロセッサ時分割システムにおける負荷分散アルゴリズム”（電子情報通信学会論文誌 D-I Vol.J72-D-I No.2 PP.75-82 1989年2月）に記載されているように、２つのプロセッサモジュール間でCPU利用率あるいは応答時間のようなシステム稼働情報により、プロセッサモジュール間の負荷分散を行っていた。このシステムでは、マスタプロセッサを設けていないため、２つのプロセッサモジュール間で負荷分散は可能であるが、システム全体を考慮した負荷分散は行えないという問題があった。

【００１２】ところで、従来の負荷分散方法では、マスタプロセッサを設けておらず、通信プロセッサとトランザクションプロセッサ間、およびトランザクションプロセッサとファイルプロセッサ間でで負荷分散を行っており、システム全体での負荷分散を行っていなかった。

【００１３】最後に、並列計算機でも同様の技術が行われており、「並列計算機構成論」、富田真治著（p.195～p.200）では、以下のように記載されている。並列計算機では、フェールセーフシステムの適用により、1プロセッサに複数の機能を持たせない。プロセッサ障害時には、処理を引き継ぐのではなく、障害プロセッサを閉塞させている。SMS-201の場合には、マスタプロセッサを設けているが、負荷分散制御は行っていない。また、各プロセッサモジュールは、単一の機能のみを実行している。その結果、並列計算機では、プロセッサモジュール障害が発生すると、特定のプロセスの処理能力が低下し、システム全体の処理能力が低下する。更に、待機プロセスを設けていないため、予備のルートを設定することができなくなり、障害時におけるシステム回復時間が長くなるという問題がある。

【００１４】また、プロセッサモジュールで障害が発生すると、チェックポイントデータの取得により、実行中の電文の引き継ぎは可能であるが、入力待ち行列に格納されている電文は消滅する。このため、電文の再送処理が必要となり、回復時間が長くなるという問題が残る。

【００１５】本発明の目的は、クラスタ型計算機システムにおいて、通信プロセッサモジュールが他のすべてのプロセッサモジュールを管理し、プロセッサモジュール間の負荷を均一にし、システム全体の処理能力を向上させることにある。また、本発明の他の目的は、受信側プロセッサモジュールの処理時間により予備のプロセッサモジュールを選んで電文を送信させておき、受信側プロセッサモジュールで障害が発生しても、電文の再送処理をなくし、回復時間を短縮させることにある。

【００１６】

【課題を解決するための手段】上記のようにプロセスを分割したシステムの処理能力は、プロセス毎に処理能力を算出し、これらのうち最も低いプロセスの処理能力になる。このため、同一プロセスを実行しているプロセッサモジュール間に負荷のばらつきがあると、システム処理能力の低下の要因となる。

【００１７】本発明の目的を達成するために、すべての

プロセッサモジュールを制御用通信パスとプロセス間通信パスにより接続し、受信側プロセッサモジュール（以下、トランザクションプロセッサモジュールとファイルプロセッサモジュール）は、電文の実行時間等のシステム稼働情報を制御用通信パスを介して、通信プロセッサモジュールに送信する。一方、通信プロセッサモジュールはトランザクションプロセッサモジュールとファイルプロセッサモジュールの処理時間を算出し、処理時間が最小となるトランザクションプロセッサモジュールとファイルプロセッサモジュールを選択する。これにより、通信プロセッサモジュールはこれらの受信側プロセッサモジュールに電文を送信することにより、プロセッサモジュールの負荷を均一にし、システム全体の処理能力を向上させる。

【００１８】本発明の他の目的を達成するために、上記で算出した処理時間から、トランザクションプロセッサモジュールとファイルプロセッサモジュールはそれぞれ予備のプロセッサモジュールを選び、予備のプロセッサモジュールにも電文を送信し、実際に電文処理を行っているプロセッサモジュールに障害が発生しても、回復時間を短縮させることができる。

【００１９】電文の受信側プロセッサモジュールは、制御用通信パスを介して、一定間隔毎に通信プロセッサモジュールにシステム稼働情報として、待ち行列に格納されている電文のタイプとその個数を送信する。一方、受信側プロセッサモジュールは、システム立ち上げ時に、通信プロセッサモジュールに処理能力や送信する電文の処理ステップ数の情報を送信しておく。そして、通信プロセッサモジュールは、電文を送信する場合に受信側プロセッサモジュールの処理時間を計算する。そして、処理時間の最も短いトランザクションプロセッサモジュールとファイルプロセッサモジュールとこれらの予備のプロセッサモジュールを選び、これらのトランザクションプロセッサモジュールとファイルプロセッサモジュールに電文を送信する。障害が発生すると、予備のプロセッサモジュールが引き継ぎ処理を実行する。

【００２０】本発明では、プロセッサモジュールは、プロセッサ、メモリ、IOPおよびIO制御装置から構成する。複数のプロセッサモジュールは、プロセス間通信パスと制御用通信パスにより接続される。電文処理の機能は、端末との通信を行う通信プロセス、トランザクションプロセスおよびディスクとのアクセスを制御するファイルプロセスに分割される。

【００２１】まず、トランザクションプロセッサモジュールおよびファイルプロセッサモジュールは、制御用通信パスを介して、待ち行列に格納されている電文のタイプとその個数をシステム稼働情報として、通信プロセッサモジュールに送信する。

【００２２】一方、通信プロセッサモジュールは、受信した待ち行列に格納されている電文のタイプとその個数、およびあらかじめ受信側プロセッサモジュールから受信しておいた受信側プロセッサモジュールの処理能力や電文の処理ステップ数の情報から処理時間を計算する。そして、通信プロセッサモジュールは処理時間が最も短いプロセッサモジュールを選び、選んだ受信側プロセッサモジュールに電文を送信する。同様に、予備となるプロセッサモジュールを選び、これらのプロセッサモジュールに電文を送信しておく。

【００２３】このようにして、通信プロセッサモジュールは、受信側プロセッサモジュールから得た情報に基づいて、各受信側プロセッサモジュールの負荷を均一とし、システム全体の処理能力を向上できる。また、プロセッサモジュールで障害が発生しても、予備のプロセッサモジュールが電文を受信しているため、回復時間を短縮できる。

【００２４】

【発明の実施の形態】以下、本発明の実施例を示す。図１は、本発明によるシステム構成図である。本発明によるシステム構成は、９個のプロセッサモジュール（１１～１９）を前提とする。すべてのプロセッサモジュール（１１～１９）は、制御用通信パス（１）およびプロセス間通信パス（３）により接続する。

【００２５】制御用通信パス（１）は、システム稼働情報、リセットメッセージおよびaliveメッセージを通信する。リセットメッセージは障害の発生したプロセッサモジュール（１１～１９）へ閉塞を要求するメッセージである。aliveメッセージはプロセッサモジュール（１１～１９）が正常に動作しているかどうかチェックするメッセージである。システム稼働情報は、本発明で使用するメッセージであり、プロセッサモジュール（１１～１９）の実行中の情報を示すものである。

【００２６】プロセス間通信パス（３）は、プロセス間で電文を通信するために使用する。プロセッサモジュール（１１～１３）は、図５で説明するように通信プロセス（３０）を搭載する。このため、プロセッサモジュール（１１～１３）は業務用LAN（４）および回線切替装置（５）と接続し、端末（６－１～４）と通信できるようにする。また、プロセッサモジュール（１１～１３）は受信電文の識別が必須であるため、電文に時刻印を付与する。このために、プロセッサモジュール（１１～１３）はタイマ（７）と接続されている。プロセッサモジュール（１１～１３）は、共有ディスク（２－０）を共有し、障害発生時に、処理を引き継ぐために必須なチェックポイント情報を格納する。

【００２７】一方、他のプロセッサモジュール（１４～１９）は、通信プロセス（３０）を搭載しないため、業務用LAN（４）あるいは回線切替装置（５）と接続されていない。プロセッサモジュール（１４～１６）は共有ディスク（２－１）を共有し、プロセッサモジュール（１４～１６）の障害発生時に、処理を引き継ぐために

必須なチェックポイント情報を格納する。プロセッサモジュール（１７～１９）は共有ディスク（２－２）を共有し、プロセッサモジュール（１７～１９）の障害発生時に、処理を引き継ぐために必須なチェックポイント情報を格納する。また、共有ディスク（２－２）には、端末（６）が参照あるいは更新するデータを格納する。

【００２８】図2は本発明の処理概要を示す図である。トランザクションプロセッサモジュール（１４～１６）は、一定間隔毎にシステム稼働情報を通信プロセッサモジュール（１１～１３）に送信する（処理　５０）。同様に、ファイルプロセッサモジュール（１７～１９）は、一定間隔毎にシステム稼働情報を通信プロセッサモジュール（１１～１３）に送信する（処理　５１）。

【００２９】一方、通信プロセッサモジュール（１１～１３）は、トランザクションプロセッサモジュール（１４～１６）とファイルプロセッサモジュール（１７～１９）のすべての組合せにおいて、処理時間を算出し、これが最小となるように電文を送信するトランザクションプロセッサモジュール（１５）とファイルプロセッサモジュール（１９）を決定する。

【００３０】次に、プロセッサモジュール（１１～１９）の処理手順について説明する。通信プロセッサモジュール（１２）は端末（６）から電文を受信する（処理　６０）。そして、通信プロセッサモジュール（１２）は、処理時間が最小となるトランザクションプロセッサモジュール（１５）に電文を送信する。更に、トランザクションプロセッサモジュール（１５）は、通信プロセッサモジュール（１２）から受信した処理時間が最小となるファイルプロセッサモジュール（１９）に電文を送信する（処理　６２）。

【００３１】そして、ファイルプロセッサモジュール（１９）はディスクの読み出し処理（処理　６３）および書き込み処理を実行する（処理　６４）。そして、その処理結果をトランザクションプロセッサモジュール（１５）に戻す（処理　６５）。更に、トランザクションプロセッサモジュール（１５）は、処理結果を通信プロセッサモジュール（１２）に戻す（処理　６６）。最後に、通信プロセッサモジュール（１２）は処理結果を端末（６）に戻す（処理　６７）。

【００３２】このように、通信プロセッサモジュール（１２）は、受信側プロセッサモジュール（１４～１９）から常に電文のタイプとその個数を受信しておき、通信プロセッサモジュール（１１～１３）は、受信側プロセッサモジュール（１４～１９）の処理能力および電文の処理ステップ数により、処理時間を計算し、処理時間が最小となるプロセッサモジュール（１４～１９）に電文を送信する。これにより、通信プロセッサモジュール（１２）は受信側プロセッサモジュール（１４～１９）の負荷を均一にし、システム全体の処理能力を向上させることが可能となる。

【００３３】更に、通信プロセッサモジュール（１２）は処理時間により、予備のプロセッサモジュール（１６、１７）を選び、予備のプロセッサモジュール（１６、１７）は電文受信処理（処理　７０～７２）のみ実行することにより、障害時に電文の再送処理をなくすことが可能となり、回復時間を短縮できる。

【００３４】図3は、プロセッサモジュールの構成図である。プロセッサモジュール（１１～１９）は業務用LAN（４）あるいは回線切替装置（５）との接続を除いて、同一構成であるため、プロセッサモジュール（１１）を例にして説明する。プロセッサモジュール（１１）はプロセッサ（１１－１）、メモリ（１１－２）、IOP（１１－３）、回線制御装置（１１－４）、プロセス間通信パス制御装置（１１－５）、ディスク制御装置（１１－６）、制御用通信パス制御装置（１１－７）、業務用LAN制御装置（１１－８）、およびシステム監視装置（１１－９）から構成される。プロセッサモジュール（１２、１３）は、プロセッサモジュール（１１）と同様に通信プロセス（３０）を搭載するため、同一の構成である。なお、回線制御装置（１１－４）、プロセス間通信パス制御装置（１１－５）、ディスク制御装置（１１－６）、制御用通信パス制御装置（１１－７）、および業務用LAN制御装置（１１－８）を合わせて、IO制御装置（１１－１０）という。

【００３５】一方、プロセッサモジュール（１４～１９）は、業務用LAN（４）あるいは回線切替装置（５）とは接続されないため、これらの制御装置は不要となる。プロセッサモジュール（１４～１９）は、プロセッサ（１４－１～１９－１）、メモリ（１４－２～１９－２）、IOP（１４－３～１９－３）、プロセス間通信パス制御装置（１４－５～１９－５）、ディスク制御装置（１４－６～１９－６）、制御用通信パス制御装置（１４－７～１９－７）、およびシステム監視装置（１４－９～１９－９）から構成される。

【００３６】図4は電文処理の概要を示す図である。電文処理は、端末（６）から電文を受信する。そして、ディスク装置（２－２）から必要なデータを読み出す。電文処理を実行し、処理結果をディスク装置（２－２）に書き込む。そして、端末（６）に応答を返す。このため、本実施例では、電文処理は、端末（６）との通信を行う通信プロセス（３０）、トランザクションプロセス（３１）、およびディスク装置（２－２）のアクセスを制御するファイルプロセス（３２）に分割される。

【００３７】図5は、ソフトウェアの構成と電文処理の分割を示す図である。プロセッサモジュール（１１～１９）のソフトウェアは、それぞれモニタ（１１－２０～１９－２０）、OS（１１－２１～１９－２１）、およびプロセス（３０～３２）から構成される。モニタ（１１－２０～１９－２０）は、プロセッサモジュール間通信情報を管理する。

【００３８】電文処理の機能は、通信プロセス（３０）、トランザクションプロセス（３１）、およびファイルプロセス（３２）に分割される。そして、通信プロセス（３０）はプロセッサモジュール（１１～１３）に、トランザクションプロセス（３１）はプロセッサモジュール（１４～１６）に、ファイルプロセス（３２）はプロセッサモジュール（１７～１９）にそれぞれ搭載される。

【００３９】以下、トランザクションプロセッサモジュール（１４～１６）とファイルプロセッサモジュール（１７～１９）を合わせて、受信側プロセッサモジュール（１４～１９）という。

【００４０】図６は、IO制御装置のシステム構成図である。ここでは、回線制御装置（１１－４）を例にして、説明する。回線制御装置（１１－４）は、プロセッサ（１１－４－１）、メモリ（１１－４－２）、バッファ（１１－４－３）、ディスク制御部（１１－４－４）で構成される。バッファ（１１－４－３）には、他のプロセッサモジュール（１４～１６）から受信した電文を格納する入力待ち行列として、入力待ち行列２０（１１－４－５）と入力待ち行列２１（１１－４－７）を設け、また他のプロセッサモジュール（１４～１６）に送信する電文を格納する出力待ち行列として、出力待ち行列２０（１１－４－６）と出力待ち行列２１（１１－４－８）を設ける。

【００４１】ここで、端末（６）からディスク（２－２）への情報の流れに対応して、上位のプロセッサモジュールおよび下位のプロセッサモジュールを、以下のように定義する。

【００４２】自身のプロセッサモジュールからディスク（２－２）側のプロセッサモジュールを下位のプロセッサモジュールとする。自身のプロセッサモジュールから端末（６）側のプロセッサモジュールを上位のプロセッサモジュールとする。例えば、トランザクションプロセッサモジュール（１４～１６）から見ると、下位のプロセッサモジュールはファイルプロセッサモジュール（１７～１９）であり、上位のプロセッサモジュールは通信プロセッサモジュール（１１～１３）である。

【００４３】IO制御装置（１１－１０）の待ち行列について、入力待ち行列２０（１１－４－５）と出力待ち行列２０（１１－４－６）は上位のプロセッサモジュール用であり、入力待ち行列２１（１１－４－７）と出力待ち行列２１（１１－４－８）は下位のプロセッサモジュール用である。

【００４４】IO制御装置（１１－１０）では４つの待ち行列があるが、システム稼働情報は、上位および下位のプロセッサモジュールから受信している電文の情報であるため、入力待ち行列２０（１１－４－５）と入力待ち行列２１（１１－４－７）を対象とする。

【００４５】図７はプロセッサモジュールの状態遷移図である。ここでは、現用状態（１５０）、輻輳状態（１５１）、およびオフライン状態（１５２）の３つの状態を設ける。現用状態（１５０）とは、正常に電文処理を実行している状態である。輻輳状態（１５１）は、一時的に処理能力の以上の電文を受信し、他のプロセッサモジュール（１１～１９）からの電文受信を禁止している状態である。オフライン状態（１５２）は障害が発生している状態である。

【００４６】現用状態（１５０）で処理を実行中に、処理時間が一定値以上になると、輻輳状態（１５１）に遷移する（１５４）。輻輳状態（１５１）の場合に、処理時間が一定値以下になると、現用状態（１５０）に遷移する（１５５）。現用状態（１５０）あるいは輻輳状態（１５１）で障害が発生すると、オフライン状態（１５２）に遷移する（１５６、１５７）。その後、障害から回復すると、オフライン状態（１５２）から現用状態（１５０）に遷移する（１５８）。

【００４７】図８は、１６進コードで表わしたプロセッサモジュールの状態コードを示す図である。現用状態（１５０）は（０１）16と、輻輳状態（１５１）は（０２）16と、オフライン状態（１５２）は（０３）16とする。

【００４８】図９は制御用通信パスにおける電文フォーマットを示す図である。電文フォーマットは、送信元アドレス（８０）、送信先アドレス（８１）、制御用電文種別（８２）、システム稼働情報１（８３）、およびシステム稼働情報２（８４）から構成される。

【００４９】以下、制御用通信パス（１）における電文情報について詳細に説明する。図１０は、１６進コードで表わした制御用電文種別コードを示す図である。制御用電文（８２）には、リセットメッセージ、aliveメッセージ、およびシステム稼働情報がある。これらのコードは図１０に示すように、それぞれ（１０）16、（０１）16および（０２）16とする。

【００５０】図１１は、システム稼働情報１とシステム稼働情報２の組合せと、対応する意味を示す図である。システム稼働情報１（８３）が（０１）16の場合には、入力待ち行列２０と入力待ち行列２１に格納されている電文のタイプとその個数を示す。

【００５１】システム稼働情報１（８３）が（１０）16の場合には、プロセッサモジュール（１１～１９）あるいはコンソール（８）に転送するイベント情報を示す。さらに、システム稼働情報２（８４）が（０１）16の場合には１つのプロセッサモジュールで輻輳発生を、（０２）16の場合には同一プロセスの全プロセッサモジュールで輻輳発生を、（０３）16の場合には障害発生を、（１０）16の場合には輻輳から回復したことを、（１１）16の場合には障害から回復したことをそれぞれ示す。

【００５２】図１２は電文のタイプとその個数を示す図である。入力待ち行列２０に格納されている電文は、電

文タイプ（８４－１－０～８４－５－０）とその個数（８４－１－１～８４－１－５）であり、入力待ち行列２１に格納されている電文は、電文タイプ（８４－１－２～８４－５－２）とその個数（８４－１－３～８４－１－３）である。

【００５３】図１３は電文のタイプとそのコードの対応を示す図である。ここでは、電文１は（０１)16とし、以下順にコードを設定し、最後の電文５は（０５)16とする。通常のデータは、図１３に示した電文の授受の後に送受信される。従って、以下の説明では、プロセスの中での通常のデータの送受信については省略する。

【００５４】図１４はプロセッサモジュールとコンソールのそれぞれのアドレスを示す図である。このアドレスは、プロセス間通信パス（３）と制御用通信パス（１）ともに、同じアドレスを使用する。プロセッサモジュール（１１）のアドレスは（００)16とする。以下、プロセッサモジュール（１２～１９）とコンソール（８）のアドレスを図１４に示す通りに設定する。

【００５５】図１５はプロセス間通信の電文フォーマットを示す図である。電文フォーマットは、送信元アドレス（９０）、送信先アドレス（９１）、電文本体（９２）、下位のプロセッサモジュールのアドレス（９３）、および識別子（９４）から構成される。

【００５６】図１６はトランザクションプロセッサモジュールの入力待ち行列に格納されている電文を示す図である。３つのトランザクションプロセッサモジュール（１４～１６）の入力待ち行列２０（１４－４－５～１６－４－５）には、それぞれ５個、２個、３個の電文が格納されており、入力待ち行列２１（１４－４－７～１６－４－７）にはそれぞれ１個、０個、０個の電文が格納されてる。これらの電文のタイプは図１６に示す通りである。

【００５７】図１７はファイルプロセッサモジュールの入力待ち行列に格納されている電文を示す図である。ファイルプロセッサモジュール（１７～１９）の入力待ち行列２０（１７－４－５～１９－４－５）には、それぞれ３個、０個、２個の電文が格納されている。電文のタイプは図１７に示す通りである。ファイルプロセッサモジュールは最下位のプロセッサモジュールであるため、入力待ち行列２１（１７－４－７～１９－４－７）は使用しない。

【００５８】図１８は、トランザクションプロセッサモジュールの電文毎の処理ステップ数と、ファイルプロセッサモジュールとの通信回数を示す図である。ここでは、電文１から電文５におけるトランザクションプロセッサモジュール（１４～１６）の実行ステップ数（上位のプロセッサモジュール用のST1(i)と下位のプロセッサモジュール用のST2(i)）と、ファイルプロセッサモジュール（１７～１９）および通信プロセッサモジュール（１１～１３）との通信回数（CT1(i)とCT2(i)）を示す。

【００５９】これらの情報は、すべての通信プロセッサモジュール（１１～１３）が所有し、処理時間を算出するために使用する。ここでは、通信プロセッサモジュール（１１～１３）およびファイルプロセッサモジュール（１７～１９）から受信した電文毎の処理ステップ数を示す。

【００６０】図１９はファイルプロセッサモジュールの電文毎の処理ステップ数とIO発行回数を示す図である。ここでは、電文１～電文５におけるファイルプロセッサモジュール（１７～１９）の実行ステップ数（SF1(i)）、通信回数(CF(i))、およびIO回数（IF1(i)）を示す。これらの情報は、すべての通信プロセッサモジュール（１１～１３）が所有し、処理時間を算出するために使用する。

【００６１】図２０は、各トランザクションプロセッサモジュールの処理能力と状態を示す図である。図２１は、各ファイルプロセッサモジュールの処理能力と状態を示す図である。

【００６２】図２０と図２１について、処理能力は１秒当りの実行ステップ数を表わす。また、プロセッサモジュールの状態は図７に示したものである。ここでは、プロセッサモジュール（１４）は輻輳状態（１５１）とし、プロセッサモジュール（１８）は障害が発生しオフライン状態（１５２）とする。他のプロセッサモジュール（１５～１７、１９）は正常に動作しており、現用状態（１５０）とする。

【００６３】これ以降、各プロセッサモジュール（１１～１９）の処理手順について説明する。最初に、図２２から図２４より、システム稼働情報の送受信処理について説明する。

【００６４】図２２はシステム稼働情報による処理手順を示す図である。図２３は通信プロセッサモジュールの処理時間の算出式である。この算出式より、トランザクションプロセッサモジュール（１４～１６）の処理時間を算出する。トランザクションプロセッサモジュール（１４）は図２０より輻輳状態（１５１）である。処理時間は、トランザクションプロセッサモジュール（１５）が最小となり、トランザクションプロセッサモジュール（１６）が２番目に小さい。トランザクションプロセッサモジュール（１５）がトランザクションプロセスの処理を実行し、トランザクションプロセッサモジュール（１６）を予備のプロセッサモジュールとする。

【００６５】同様に、ファイルプロセッサモジュール（１７～１９）の処理時間を算出する。ファイルプロセッサモジュール（１８）は図２１よりオフライン状態である。処理時間は、ファイルプロセッサモジュール（１９）が最小となり、ファイルプロセッサモジュール（１７）が２番目に小さい。ファイルプロセッサモジュール（１９）がトランザクションプロセスの処理を実行し、

ファイルプロセッサモジュール（１７）を予備のプロセッサモジュールとする。

【００６６】図２４は制御用電文の一例を示す図である。図２４（１）は待ち行列に格納されている電文の情報の転送を示す。図２４（２）は、上位のプロセッサモジュールへの電文送信禁止の通知を示す。図２４（３）、（４）、（５）は、それぞれケース１からケース３のコンソール（８）への表示を示す。

【００６７】ここでは、ファイルプロセッサモジュール（１９）が通信プロセッサモジュール（１２）にシステム稼働情報を送信する場合を示す。まず、ファイルプロセッサモジュール（１９）は、通信プロセッサモジュール（１２）に対して、システム稼働情報として、入力待ち行列２０（１９－４－７）に格納されている電文のタイプとその個数を送信する（処理　４００）。

【００６８】この電文を図２４（１）に示す。ここでは、ファイルプロセッサモジュールはプロセッサモジュール（１９）とする。送信元アドレス（８０）はプロセッサモジュール（１９）であるため（２２）16を、送信先アドレス（８１）はプロセッサモジュール（１２）であるため（０１）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）は電文のタイプと個数であるため（０１）16を、それぞれ設定する。また、システム稼働情報２（８４）は電文のタイプ毎にその個数を格納するため、図１７に示している通り、ファイルプロセッサモジュール（１９）には電文５が１個と電文２が１個格納されているため、電文のタイプ毎に、電文のタイプの順に、電文のタイプとその個数を、電文１では０個であるため、（０１００）16、電文２では、（０２０１）16、電文３では（０３００）16、電文４では（０４００）16を、電文５では（０５０１）16を設定する。ファイルプロセッサモジュール（１９）は入力待ち行列２１（１９）を使用しないため、入力待ち行列２１用（８４－１－２～８４－５－３）には（FF）16を格納する。

【００６９】通信プロセッサモジュール（１２）は、ファイルプロセッサモジュール（１９）の入力待ち行列２０（１９－４－５）と入力待ち行列２１（１９－４－７）に格納されている電文タイプとその個数を受信する（処理　４０１）。そして、通信プロセッサモジュール（１２）は、ファイルプロセッサモジュール（１９）の処理時間を算出する（処理　４０２）。通信プロセッサモジュールの処理時間の算出式を図２３に示す。

【００７０】ここで、通信プロセッサモジュール（１２）は、受信側プロセッサモジュール（１４～１９）の電文の実行ステップ数、通信回数、およびIO待ち時間を事前に格納しておく。この例を図１８と図１９に示す。さらに、受信側プロセッサモジュール（１４～１９）の処理能力とプロセッサモジュールの状態を事前に格納しておく。この例を図２０と図２１に示す。

【００７１】図２３は処理時間の算出式である。トランザクションプロセッサモジュール（１４～１６）の処理時間（RT(j)）は、電文タイプ毎の実行ステップ数（ST1(i)とST2(i)）と電文数（NT1(i)とNT2(i)）、およびプロセス間通信回数（CT1(i)とCT2(i)）とプロセス間通信のステップ数（PS）から算出する。

【００７２】ファイルプロセッサモジュール（１７～１９）の処理時間（RF(j)）は、電文タイプ毎の実行ステップ数（SF1(i)）と電文数（NF1(i)）、プロセス間通信回数（CF1(i)）とプロセス間通信のステップ数（PS）、およびIO発行回数（IF1(i)）とIO待ち時間（IT）から算出する。このように、通信プロセッサモジュール（１２）は図２３の算出式を用いて、受信側プロセッサモジュール（１４～１９）の処理時間（RT(j),RF(j)）を計算する。

【００７３】そして、通信プロセッサモジュール（１２）は、すべての受信側プロセッサモジュール（１４～１９）の処理時間が一定値（L１）以上かどうかチェックする（処理　４０３）。そして、これが一定値以上（L１）ならば、このプロセッサモジュールは輻輳状態（１５１）と判定し、このプロセッサモジュールへの電文送信を禁止する（処理　４０４）。

【００７４】さらに、すべてのファイルプロセッサモジュール（１７～１９）が輻輳状態（１５１）かどうか判定する（処理　４０５）。すべてのファイルプロセッサモジュール（１７～１９）が輻輳状態（１５１）ならば、トランザクションプロセッサモジュール（１１～１３）に対して電文送信禁止を通知する（処理　４０６）。ここで、この通知する電文を図２４（２）に示す。ここでは通信プロセッサモジュール（１２）に送信する場合を示す。送信元アドレス（８０）はプロセッサモジュール（１５）であるため（１１）16を、送信先アドレス（８１）はプロセッサモジュール（１１）であるため（００）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）はイベント情報であるため（１０）16を、システム稼働情報２（８４）は輻輳発生であるため（０１）16をそれぞれ設定する。

【００７５】また、受信側プロセッサモジュール（１４～１９）が輻輳状態（１５１）であるかを判定し（処理　４０７）、輻輳状態（１５１）のプロセッサモジュールの処理時間が一定値（L２）以下になっている（処理　４０８）と、下位プロセッサモジュールへの電文送信を再開し、上位のプロセッサモジュールに対して電文送信再開を通知する（処理　４０９）。

【００７６】そして、ケース１～３の場合には、コンソール（８）に通知する（処理　４１０）。ここで、ケース１の通知する電文を図２４（３）に示す。送信元アドレス（８０）はプロセッサモジュール（１２）であるため（０１）16を、送信先アドレス（８１）はコンソール

（８）であるため（３０）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）はイベント情報であるため（１０）16を、システム稼働情報２（８４）は１つのプロセッサモジュールで輻輳発生であるため（０１）16をそれぞれ設定する。

【００７７】ここで、ケース２の通知する電文を図２４（４）に示す。送信元アドレス（８０）、送信先アドレス（８１）、制御用電文種別（８２）、システム稼働情報１（８３）はケース１と同一である。システム稼働情報２（８４）は、すべてのファイルプロセッサモジュール（17～１９）で輻輳発生であるため、（０２）16を設定する。

【００７８】ここで、ケース３の通知する電文を図２４（５）に示す。送信元アドレス（８０）、送信先アドレス（８１）、制御用電文種別（８２）、システム稼働情報１（８３）はケース１と同一であり、システム稼働情報２（８４）は、ファイルプロセッサモジュール（１７～１９）は輻輳解除であるため、（１０）16を設定する。このようにして、受信側プロセッサモジュール（１４～１９）は、通信プロセッサモジュール（１２）あるいはコンソール（８）に輻輳発生および輻輳解除を通知できる。

【００７９】次に、図２５から図２７より、障害時および障害回復時の処理手順について説明する。図２５は障害時の処理手順を示す図である。図２７に障害時の制御用電文を示す。

【００８０】ファイルプロセッサモジュール（１９）は、一定間隔毎に常にファイルプロセッサモジュール（１７）にaliveメッセージを送信している（処理　５００）。ファイルプロセッサモジュール（１７）は、aliveメッセージの受信によりファイルプロセッサモジュール（１９）が正常に動作していると認識している。ファイルプロセッサモジュール（１９）で障害が発生すると（処理　５０１）、ファイルプロセッサモジュール（１７）はaliveメッセージの途絶により、ファイルプロセッサモジュール（１９）の障害を検出する（処理　５０２）。

【００８１】ここで、正常動作を通知する電文を図２７（１）に示す。ファイルプロセッサモジュール（１９）がファイルプロセッサモジュール（１７）にリセットメッセージを送信する場合を示す。送信元アドレス（８０）はファイルプロセッサモジュール（１９）であるため（２２）16を、送信先アドレス（８１）はファイルプロセッサモジュール（１７）であるため（２０）16を、制御用電文種別（８２）はaliveメッセージであるため（０１）16を、システム稼働情報１（８３）とシステム稼働情報２（８４）は使用しないため（FF）16を、それぞれ設定する。

【００８２】更に、ファイルプロセッサモジュール（１７）は障害の発生したファイルプロセッサモジュール（１９）にリセットを要求する（処理　５０３）。そして、ファイルプロセッサモジュール（１９）はモニタ（１９－２０）がIO制御装置（１９－１０）をリセットする処理を実行する（処理　５０４）。

【００８３】ここで、リセット要求を通知する電文を図２７（２）に示す。ファイルプロセッサモジュール（１８）がファイルプロセッサモジュール（１９）にリセットメッセージを送信する場合を示す。送信元アドレス（８０）はファイルプロセッサモジュール（１７）であるため（２０）16を、送信先アドレス（８１）はファイルプロセッサモジュール（１９）であるため（２２）16を、制御用電文種別（８２）はリセットメッセージであるため（１０）16を、システム稼働情報１（８３）とシステム稼働情報２（８４）は使用しないため（FF）16をそれぞれ設定する。

【００８４】更に、ファイルプロセッサモジュール（１７）は障害発生をコンソール（８）に通知し（処理　５０５）、通信プロセッサモジュール（１２）に通知する（処理　５０６）。そして、通信プロセッサモジュール（１２）はこれを受信する（処理　５０７）。

【００８５】ここで、コンソールに障害発生を通知する電文を図２７（３）に示す。プロセッサモジュール（１７）がコンソール（８）に障害発生を送信する場合を示す。送信元アドレス（８０）はファイルプロセッサモジュール（１７）であるため（２０）16を、送信先アドレス（８１）はコンソール（８）であるため（３０）１６を、制御用電文種別（８２）はシステム稼働情報であるため（０２）１６を、システム稼働情報１（８３）はイベント情報であるため（１０）16、システム稼働情報２（８４）は障害発生通知であるため（０３）16をそれぞれ設定する。

【００８６】次に、通信プロセッサモジュールに障害発生を通知する電文を図２７（４）に示す。ファイルプロセッサモジュール（１７）が通信プロセッサモジュール（12）に障害発生を送信する場合を示す。送信元アドレス（８０）はファイルプロセッサモジュール（１７）であるため（２０）16を、送信先アドレス（８１）は通信プロセッサモジュール（１２）であるため（０１）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）はプロセッサモジュール情報であるため（１０）16、システム稼働情報２（８４）は障害発生通知であるため（０３）16をそれぞれ設定する。

【００８７】図２６は障害回復時の処理手順を示す図である。図２７（５）は障害回復時の制御用電文を示す図である。障害の発生したファイルプロセッサモジュール（１９）は障害から回復したことを通信プロセッサモジュール（１２）に通知する（処理　５５０）。そして、通信プロセッサモジュール（１２）は障害回復の通知を

受信する（処理　５５１）。

【００８８】図２７（５）は通信プロセッサモジュールに障害回復を通知する電文である。ファイルプロセッサモジュール（１９）が通信プロセッサモジュール（１２）に障害回復を送信する場合を示す。送信元アドレス（８０）はプロセッサモジュール（１９）であるため（２２）16を、送信先アドレス（８１）は通信プロセッサモジュール（１２）であるため（０１）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）はイベント情報であるため（１０）16、システム稼働情報２（８４）は障害回復通知であるため（１１）16をそれぞれ設定する。

【００８９】最後に、障害回復の通知をコンソール（８）に表示する（処理　５５２）。次に、コンソールに障害回復を通知する電文を図２７（６）に示す。通信プロセッサモジュール（１２）がコンソール（８）に障害回復を送信する場合を示す。送信元アドレス（８０）は通信プロセッサモジュール（１２）であるため（０１）16を、送信先アドレス（８１）はコンソール（８）であるため（３０）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）はイベント情報であるため（１０）16、システム稼働情報２（８４）は障害回復通知であるため（１１）16をそれぞれ設定する。

【００９０】このようにして、プロセッサモジュール（１１～１９）間あるいはプロセッサモジュール（１１～１９）とコンソール（８）間で制御用通信パス（１）を介して通信を行うことが可能となる。

【００９１】まず、図２８と図３１を用いて、３つのプロセッサモジュール（１１～１９）間での電文処理手順について説明する。図２８は電文処理手順を示す図である。図２９はプロセス間通信の電文を示す図である。通信プロセッサモジュール（１２）は、端末（６）から電文を受信すると（処理　６００）、各トランザクションプロセッサモジュール（１４～１６）の処理時間を算出し、このうち最も処理時間の短いプロセッサモジュールを選ぶ（処理　６０１）。図２８では、トランザクションプロセッサモジュール（１５）とファイルプロセッサモジュール（１９）とが選択されたとする。通信プロセッサモジュール（１２）はトランザクションプロセッサモジュール（１５）に電文を送信する。

【００９２】図２９（１）は、通信プロセッサモジュール（１２）がトランザクションプロセッサモジュール（１５）に送信する電文を示す。送信元アドレス（９０）は通信プロセッサモジュール（１２）であるため（０１）16を、送信先アドレス（９１）はトランザクションプロセッサモジュール（１５）であるため（１１）16を、電文本体（９２）は（０１０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はファイルプロセッサモジュール（１９）であるため（２２）16、電文の識別子（９４）は（０００１）16をそれぞれ設定する。

【００９３】トランザクションプロセッサモジュール（１５）はファイルプロセッサモジュール（１９）にディスク（２－２）からの読み出しを要求する（処理　６０５）。図２９（２）に読み出し要求の電文を示す。送信元アドレス（９０）はプロセッサモジュール（１５）であるため（１１）16を、送信先アドレス（９１）はファイルプロセッサモジュール（１９）であるため（２２）16を、電文本体（９２）は（０２０３０１）16を、下位のプロセッサモジュールのアドレス（９３）はないため（FF）16、電文の識別子（９４）は（０００１）16をそれぞれ設定する。

【００９４】ファイルプロセッサモジュール（１９）は読み出し要求を受信すると（処理６０６）、ディスク（２－２）の読み出し処理を実行する（処理　６０７）。そして、読み出し結果をトランザクションプロセッサモジュール（１５）に送信する（処理　６０８）。図２９（３）に読み出し結果送信の電文を示す。送信元アドレス（９０）はプロセッサモジュール（１９）であるため（２２）16を、送信先アドレス（９１）はプロセッサモジュール（１５）であるため（１１）16を、電文本体（９２）は（０４０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため（FF）16を、電文の識別子（９４）は（０００１）16を、それぞれ設定する。

【００９５】さらに、トランザクションプロセッサモジュール（１５）はファイルプロセッサモジュール（１９）にディスク（２－２）への書き込みを要求する（処理　６０９）。図２９（４）に書き込み要求の電文を示す。送信元アドレス（９０）はプロセッサモジュール（１５）であるため（１１）16を、送信先アドレス（９１）はプロセッサモジュール（１９）であるため（１１）16を、電文本体（９２）は（０５０４０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため（FF）16、電文の識別子（９４）は（０００１）16を、それぞれ設定する。

【００９６】ファイルプロセッサモジュール（１９）は書き込み要求を受信すると（処理６１０）、ディスク（２－２）の書き込み処理を実行し（処理　６１１）、書き込み完了を送信する（処理　６１２）。図２９（５）に書き込み完了の電文を示す。送信元アドレス（９０）はプロセッサモジュール（１９）であるため（２２）16を、送信先アドレス（９１）はプロセッサモジュール（１５）であるため（１１）16を、電文本体（９２）は（０７０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため（FF）16を、電文の識別子（９４）は（０００１）16を、それぞれ設定する。

【００９７】トランザクションプロセッサモジュール（１５）は通信プロセッサモジュール（１２）に処理結果を送信する（処理　６１３）。図２９（６）に処理結果送信の電文を示す。送信元アドレス（９０）はプロセッサモジュール（１５）であるため（１１）16を、送信先アドレス（９１）はプロセッサモジュール（１２）であるため（０１）16を、電文本体（９２）は（０１０５０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため（FF）16を、電文の識別子（９４）は（０００１）16を、それぞれ設定する。

【００９８】最後に、通信プロセッサモジュール（１２）は、端末（６）に処理結果を送信する（処理　６１４）。このように、プロセッサモジュール（１４～１６、１７～１９）間の負荷分散が可能となる。

【００９９】図３０はトランザクションプロセッサモジュールで障害が発生した場合の処理を示す図である。図３１はファイルプロセッサモジュールで障害が発生した場合の処理を示す図である。図３２はプロセス間通信の電文を示す図である。まず、図３０と図３２を用いて、トランザクションプロセッサモジュール（１５）で障害が発生した場合について説明する。

【０１００】トランザクションプロセッサモジュール（１５）で障害が発生とすると（処理６５０）、トランザクションプロセッサモジュール（１６）はaliveメッセージの途絶により、障害を検出する（処理　６５１）。そして、トランザクションプロセッサモジュール（１５）に対し、リセットを要求し（処理　６５２）、トランザクションプロセッサモジュール（１５）はリセット処理を実行する（処理６５３）。

【０１０１】リセット要求の制御用電文を図３２（１）に示す。送信元アドレス（８０）はプロセッサモジュール（１６）であるため（１２）16を、送信先アドレス（８１）はプロセッサモジュール（１５）であるため（１１）16を、制御用電文種別（８２）はリセットメッセージであるため（１０）16を、システム稼働情報１（８３）とシステム稼働情報２（８４）は使用しないため、（FF）16を格納する。そして、トランザクションプロセッサモジュール（１６）は、通信プロセッサモジュール（１２）に、障害発生を通知する（処理　６５４）。

【０１０２】障害通知の制御用電文を図３２（２）に示す。送信元アドレス（８０）はプロセッサモジュール（１６）であるため（１２）16を、送信先アドレス（８１）はプロセッサモジュール（１２）であるため（０１）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）はイベント情報であるため（１０）16を、システム稼働情報２（８４）は障害通知であるため（０３）16を、それぞれ格納する。

【０１０３】通信プロセッサモジュール（１２）はトランザクションプロセッサモジュール（１５）の処理時間を算出し、別のトランザクションプロセッサモジュール（１６）に電文を送信する（処理　６５５）。そして、トランザクションプロセッサモジュール（１６）は、電文を実行しているファイルプロセッサモジュールに通知する（処理　６５６）。そして、トランザクションプロセッサモジュール（１６）は、ファイルプロセッサモジュール（１９）に電文を送信する（処理　６５７）。

【０１０４】プロセス間通信の電文を図３２（３）に示す。送信元アドレス（９０）はプロセッサモジュール（１２）であるため（０１）16を、送信先アドレス（９１）はプロセッサモジュール（１５）であるため（１１）16を、電文本体（９２）は（０１０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はプロセッサモジュール（１９）であるため（２２）16、電文の識別子（９４）は（０００１）16をそれぞれ設定する。

【０１０５】トランザクションプロセッサモジュール（１５）は電文を受信すると（処理６５８）、ファイルプロセッサモジュール（１９）に読み出し要求を送信する（処理　６５９）。

【０１０６】一方、ファイルプロセッサモジュール（１９）は、電文を受信すると、チェックポイント情報に基づいて、回復処理を行う。書き込み処理が完了していれば、処理結果をトランザクションプロセッサモジュールに戻し、書き込み処理が完了していなければ、最初から電文処理を再実行する。

【０１０７】次に、図３１と図３２を用いて、ファイルプロセッサモジュール（１９）で障害が発生した場合について説明する。ファイルプロセッサモジュール（１９）で障害が発生とすると（処理　６７０）、ファイルプロセッサモジュール（１７）はaliveメッセージの途絶により、障害を検出する（処理　６７１）。そして、ファイルプロセッサモジュール（１９）に対し、リセットを要求し（処理　６７２）、ファイルプロセッサモジュール（１９）はリセット処理を実行する（処理　６７３）。

【０１０８】リセット要求の制御用電文を図３２（４）に示す。送信元アドレス（８０）はプロセッサモジュール（１７）であるため（２０）16を、送信先アドレス（８１）はプロセッサモジュール（１９）であるため（２２）16を、制御用電文種別（８２）はリセットメッセージであるため（１０）16を、システム稼働情報１（８３）とシステム稼働情報２（８４）は使用しないため、（FF）16を格納する。

【０１０９】そして、ファイルプロセッサモジュール（１７）は、通信プロセッサモジュール（１２）に、障害発生を通知する（処理　６７４）。

【０１１０】障害通知の制御用電文を図３２（５）に示す。送信元アドレス（８０）はプロセッサモジュール

（１７）であるため（２０）16を、送信先アドレス（８１）はプロセッサモジュール（１２）であるため（０１）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）はイベント情報であるため（１０）16を、システム稼働情報２（８４）は障害通知であるため（０３）16を、それぞれ格納する。

【０１１１】通信プロセッサモジュール（１２）は、ファイルプロセッサモジュール（１７）を介してファイルプロセッサモジュール（１９）の障害発生を受信する（処理６７５）。通信プロセッサモジュール（１２）はトランザクションモジュール（１４～１６）の処理時間を算出し、処理時間の最も短いプロセッサモジュール（１５）を選択する（処理　６７６）。そして、通信プロセッサモジュール（１２）は、トランザクションプロセッサモジュール（１５）に送信済みの電文を再送する（処理　６７７）。

【０１１２】プロセス間通信の電文を図３２（６）に示す。送信元アドレス（９０）はプロセッサモジュール（１２）であるため（０１）16を、送信先アドレス（９１）はプロセッサモジュール（１５）であるため（１１）16を、電文本体（９２）は（０１０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はプロセッサモジュール（１９）であるため（２２）16、電文の識別子（９４）は（０００１）16をそれぞれ設定する。トランザクションプロセッサモジュール（１５）は電文を受信すると（処理６７８）、ファイルプロセッサモジュール（１９）に読み出し要求を送信する（処理６７９）。

【０１１３】一方、ファイルプロセッサモジュール（１９）は、電文を受信すると、ジャーナルに基づいて、回復処理を行う。書き込み処理が完了していれば、処理結果をトランザクションプロセッサモジュール（１５）に戻し、書き込み処理が完了していなければ、最初から電文処理を再実行する。

【０１１４】通信プロセッサモジュール（１２）で障害が発生すると、一定時間経過しても、応答を戻らないことにより、端末（６）が通信プロセッサモジュール（１２）の障害を検出し、別の通信プロセッサモジュール（１３）に通知する。この結果、プロセッサモジュール（１５、１９）で障害が発生すると、電文の再送処理が必須となり、回復時間が長くなるという問題が残る。

【０１１５】最後に、予備のプロセッサモジュール（１６、１７、１３）を考慮し、電文を予備のプロセッサモジュール（１６、１７、１３）にも送信し、障害時の回復時間を短縮させる手順を示す。

【０１１６】図３３は予備のプロセッサモジュールを考慮した場合の処理を示す図である。図３４は予備のプロセッサモジュールを考慮した場合のプロセス間通信の電文を示す図である。

【０１１７】端末（６）は、業務用LAN（４）経由の場合には通信プロセッサモジュール（１２、１３）に電文を送信する。回線切替装置（５）経由の場合には、回線切替装置（５）は、２つの通信プロセッサモジュール（１２、１３）が電文を受信できるように制御を行い、端末（６）からの電文を通信プロセッサモジュール（１２、１３）が受信する（処理　７００、７０１）。本実施例では、プロセッサモジュール（１３）を予備の通信プロセッサモジュールとする。実際に通信処理を実行する通信プロセッサモジュール（１２）は、下位のプロセッサモジュール（１４～１９）の処理時間を計算し、処理時間の最も短いプロセッサモジュールと、予備のプロセッサモジュールを決める（処理　７０２）。そして、処理時間が最小となるプロセッサモジュールはトランザクションプロセッサモジュール（１５）とファイルプロセッサモジュール（１９）とする。更に、それぞれの予備のプロセッサモジュールをトランザクションプロセッサモジュール（１６）とファイルプロセッサモジュール（１７）とする。

【０１１８】通信プロセッサモジュール（１２）は、電文をトランザクションプロセッサモジュール（１５、１６）に送信する（処理　７０３）。このプロセス間通信の電文を図３４（１）に示す。送信元アドレス（９０）はプロセッサモジュール（１２）であるため（０１）16を、送信先アドレス（９１）はプロセッサモジュール（１５、１６）であるため（１１１２）16を、電文本体（９２）は（０１０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はプロセッサモジュール（１９、１７）であるため（２２、２０）16を、電文の識別子（９４）は（０００１）16をそれぞれ設定する。

【０１１９】トランザクションプロセッサモジュール（１５）は電文を受信すると（処理７０４）、読み出し要求をファイルプロセッサモジュール（１９、１７）に送信する（処理　７０６）。このプロセス間通信の電文を図３４（２）に示す。送信元アドレス（９０）はプロセッサモジュール（１５）であるため（１１）16を、送信先アドレス（９１）はプロセッサモジュール（１９、１７）であるため（２２２０）16を、電文本体（９２）は（０２０３０１）16を、下位のプロセッサモジュールのアドレス（９３）はないため（FF）16を、電文の識別子（９４）は（０００１）16を、それぞれ設定する。一方、トランザクションプロセッサモジュール（１６）は電文を受信するだけである（処理　７０５）。

【０１２０】ファイルプロセッサモジュール（１９）は、読み出し要求を受信すると（処理７０７）、ディスク（２－２）からの読み出し処理を実行する（処理　７０９）。ファイルプロセッサモジュール（１７）は電文を受信するだけである（処理７０８）。そして、ファイルプロセッサモジュール（１９）は、トランザクションプロセッサモジュール（１５、１６）に読み出し結果を

送信する（処理　７１０）。このプロセス間通信の電文を図３４（３）に示す。送信元アドレス（９０）はプロセッサモジュール（１９）であるため（２２）16を、送信先アドレス（９１）はプロセッサモジュール（１５、１６）であるため（１１１２）16を、電文本体（９２）は（０４０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため（FF）16を、電文の識別子（９４）は（０００１）16をそれぞれ設定する。

【０１２１】トランザクションプロセッサモジュール（１５）は、読み出し結果を受信すると（処理　７１１）、更に処理を継続させ、書き込み要求をファイルプロセッサモジュール（１９、１７）に送信する（処理　７１３）。このプロセス間通信の電文を図３４（４）に示す。送信元アドレス（９０）はプロセッサモジュール（１５）であるため（１１）16を、送信先アドレス（９１）はプロセッサモジュール（１９、１７）であるため（２２２０）16を、電文本体（９２）は（０５０４０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため（FF）16を、電文の識別子（９４）は（０００１）16をそれぞれ設定する。一方、トランザクションプロセッサモジュール（１６）は読み出し結果を受信するだけである（処理　７１２）。

【０１２２】ファイルプロセッサモジュール（１９）は書き込み処理を受信すると（処理７１４）、ディスク（２－２）への書き込み処理を実行する（処理　７１６）。そして、ファイルプロセッサモジュール（１９）は、書き込み完了をトランザクションプロセッサモジュール（１５、１６）に送信する（処理　７１７）。このプロセス間通信の電文を図３４（５）に示す。送信元アドレス（９０）はプロセッサモジュール（１９）であるため（２２）16を、送信先アドレス（９１）はプロセッサモジュール（１５、１６）であるため（１１１２）16を、電文本体（９２）は（０７０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため（FF）16を、電文の識別子（９４）は（０００１）16をそれぞれ設定する。一方、ファイルプロセッサモジュール（１７）は書き込み要求を受信するだけである（処理　７１５）。

【０１２３】トランザクションプロセッサモジュール（１５）は処理結果を受信する（処理７１９）と、通信プロセッサモジュール（１２、１３）に処理結果を送信する（処理　７２１）。このプロセス間通信の電文を図３４（６）に示す。送信元アドレス（９０）はプロセッサモジュール（１５）であるため（１１）16を、送信先アドレス（９１）はプロセッサモジュール（１１、１２）であるため（０１０２）16を、電文本体（９２）は（０１０５０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため、（FF）16を、電文の識別子（９４）は（０００１）16をそれぞれ設定する。トランザクションプロセッサモジュール（１６）は処理結果を受信するだけである（処理　７２０）。

【０１２４】最後に、通信プロセッサモジュール（１２）は、処理結果を受信すると（処理７２２）、端末（６）に送信する（処理　７２４）。通信プロセッサモジュール（１２）は処理結果を受信するだけである（処理　７２３）。

【０１２５】次に、各プロセッサモジュール（１５、１９）で障害が発生した場合について、図３５から図３７を用いて説明する。図３５は、トランザクションプロセッサモジュールで障害が発生した場合の処理を示す図である。図３７はプロセス間通信の電文を示す図である。

【０１２６】まず、トランザクションプロセッサモジュール（１５）で障害が発生した場合について、図３５と図３７を用いて説明する。図３５はトランザクションプロセッサモジュールで障害が発生した場合の処理を示す図である。

【０１２７】トランザクションプロセッサモジュール（１５、１６）は、通信プロセッサモジュール（１２）から同時に電文を受信している（処理　７０３～７０５）。トランザクションプロセッサモジュール（１５）で障害が発生すると（処理　７３０）、トランザクションプロセッサモジュール（１６）は、aliveメッセージの途絶により、トランザクションプロセッサモジュール（１５）の障害を検出する（処理　７３１）。

【０１２８】更に、トランザクションプロセッサモジュール（１６）は、障害の発生したトランザクションプロセッサモジュール（１５）にリセットを要求する（処理　７３２）。そして、トランザクションプロセッサモジュール（１５）は、モニタ（１５－２０）がIO制御装置（１５－１０）をリセットする処理を実行する（処理７３３）。

【０１２９】ここで、リセット要求を通知する電文を図３７（１）に示す。トランザクションプロセッサモジュール（１６）がトランザクションプロセッサモジュール（１５）にリセットメッセージを送信する場合を示す。送信元アドレス（８０）はファイルプロセッサモジュール（１６）であるため（１２）16を、送信先アドレス（８１）はファイルプロセッサモジュール（１５）であるため（１１）16を、制御用電文種別（８２）はリセットメッセージであるため（１０）16を、システム稼働情報１（８３）とシステム稼働情報２（８４）は使用しないため（FF）16を、それぞれ設定する。

【０１３０】更に、トランザクションプロセッサモジュール（１６）は、障害発生を通信プロセッサモジュール（１２）に通知する（処理　７３４）。そして、通信プロセッサモジュール（１２）は障害発生の通知を受信する（処理　７３５）。障害発生を通知する電文を図３７（２）に示す。トランザクションプロセッサモジュール（１６）が通信プロセッサモジュール（12）に障害発生を送信する場合を示す。送信元アドレス（８０）はト

ランザクションプロセッサモジュール（１６）であるため（１２）16を、送信先アドレス（８１）は通信プロセッサモジュール（１２）であるため（０１）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）はイベント情報であるため（１０）16、システム稼働情報２（８４）は障害発生通知であるため（０３）16をそれぞれ設定する。

【０１３１】そして、トランザクションプロセッサモジュール（１６）は、再度読み出し要求をファイルプロセッサモジュール（１９、１７）に送信する（処理　７３６）。読み出し要求を通知する電文を図３７（３）に示す。送信元アドレス（９０）はトランザクションプロセッサモジュール（１６）であるため（１２）16を、送信先アドレス（９１）はトランザクションプロセッサモジュール（１９、１７）であるため（２２２０）16を、電文本体（９２）は（０１０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため、（FF）16、電文の識別子（９４）は（０００１）16をそれぞれ設定する。

【０１３２】次に、ファイルプロセッサモジュール（１５）で障害が発生した場合について、図３６と図３７を用いて説明する。図３６はファイルプロセッサモジュールで障害が発生した場合の処理を示す図である。

【０１３３】ファイルプロセッサモジュール（１９、１７）は、トランザクションプロセッサモジュール（１５）から同時に電文を受信している（処理　７０６～７０８）。ファイルプロセッサモジュール（１９）で障害が発生すると（処理　７６０）、ファイルプロセッサモジュール（１７）はaliveメッセージの途絶により、ファイルプロセッサモジュール（１９）の障害を検出する（処理　７６１）。

【０１３４】更に、ファイルプロセッサモジュール（１７）は、障害の発生したファイルプロセッサモジュール（１９）にリセットを要求する（処理　７６２）。そして、ファイルプロセッサモジュール（１９）はモニタ（１９－２０）がIO制御装置（１９－１０）をリセットする処理を実行する（処理　７６３）。

【０１３５】ここで、リセット要求を通知する電文を図３７（４）に示す。ファイルプロセッサモジュール（１７）がファイルプロセッサモジュール（１９）にリセットメッセージを送信する場合を示す。送信元アドレス（８０）はファイルプロセッサモジュール（１７）であるため（２０）16を、送信先アドレス（８１）はファイルプロセッサモジュール（１９）であるため（２１）16を、制御用電文種別（８２）はリセットメッセージであるため（１０）16を、システム稼働情報１（８３）とシステム稼働情報２（８４）は使用しないため（FF）16を、それぞれ設定する。

【０１３６】更に、ファイルプロセッサモジュール（１７）は障害発生を通信プロセッサモジュール（１２）に通知する（処理　７６４）。そして、通信プロセッサモジュール（１２）はこれを受信する（処理　７６５）。障害発生を通知する電文を図３７（５）に示す。ファイルプロセッサモジュール（１７）が通信プロセッサモジュール（12）に障害発生を送信する場合を示す。送信元アドレス（８０）はファイルプロセッサモジュール（１７）であるため（２０）16を、送信先アドレス（８１）は通信プロセッサモジュール（１２）であるため（０１）16を、制御用電文種別（８２）はシステム稼働情報であるため（０２）16を、システム稼働情報１（８３）はイベント情報であるため（１０）16、システム稼働情報２（８４）は障害発生通知であるため（０３）16をそれぞれ設定する。

【０１３７】そして、ファイルプロセッサモジュール（１７）は、再度読み出し処理を実行する（処理　７６６）。そして、ファイルプロセッサモジュール（１７）は、読み出し結果をトランザクションプロセッサモジュール（１５、１６）に送信する（処理　７６７）。読み出し結果の送信を通知する電文を図３７（６）に示す。送信元アドレス（９０）はファイルプロセッサモジュール（１７）であるため（１２）16を、送信先アドレス（９１）はトランザクションプロセッサモジュール（１５、１６）であるため（１１１２）16を、電文本体（９２）は（０１０２０３）16を、下位のプロセッサモジュールのアドレス（９３）はないため、（FF）16、電文の識別子（９４）は（０００１）16をそれぞれ設定する。

【０１３８】通信プロセッサモジュール（１２）で障害が発生すると、端末（６）は両方の通信プロセッサモジュール（１２、１３）に電文を送信することにより、端末（６）への電文の再送要求をなくし、回復時間を短縮できる。本実施例では、電文処理を３つに分割し、３階層の場合について説明した。同様にして、電文処理をn個に分割したn階層のクラスタ型計算機システムにも本方式を適用できる。

【０１３９】通信プロセッサモジュール（１１～１３）（最上位のプロセッサモジュール）が、受信側プロセッサモジュールの入力待ち行列に格納されている電文情報から、受信した電文の処理時間を計算し、これが最小となるプロセッサモジュールを見つけ、これらのプロセッサモジュールで電文処理を実行することにより、各プロセスのプロセッサモジュールの負荷を均一にし、システム全体の処理能力を向上させることが可能となる。

【０１４０】更に、処理時間が２番目に短いプロセッサモジュールを予備のプロセッサモジュールとし、予備のプロセッサモジュールにも電文を送信することにより、回復時間を短縮させることが可能となる。

【０１４１】図３８は本発明によるシステム構成図（２）である。ここでは、本発明によるシステム構成は、図１のシステム構成と同様に９個のプロセッサモジ

ュール（１１～１９）を前提とする。すべてのプロセッサモジュール（１１～１９）は、制御用通信パス（１）およびプロセス間通信パス（３）により接続される。図３８では、プロセッサモジュールに２つの異なるプロセスを搭載させ、相互にバックアップさせるものである。

【０１４２】プロセッサモジュール（１１～１３）は、図５で説明した通信プロセス（３０）を搭載する。プロセッサモジュール（１１～１３）は、共有ディスク（２－０）を共有し、障害発生時に、処理を引き継ぐために必須なチェックポイント情報を格納する。

【０１４３】プロセッサモジュール（１４～１６）のそれぞれは、トランザクションプロセス（３１）とファイルプロセス（３２）を搭載する。トランザクションプロセス（３１）は現用プロセスとして稼働し、また、ファイルプロセス（３２－１）は待機プロセスとして稼働される。

【０１４４】プロセッサモジュール（１７～１９）のそれぞれは、トランザクションプロセス（３１）とファイルプロセス（３２）を搭載する。ファイルプロセス（３２）は現用プロセスとして稼働し、また、トランザクションプロセス（３１－１）は待機プロセスとして稼働される。

【０１４５】プロセッサモジュール（１４～１９）の障害を検出した際に、プロセス（３１，３２）の個数に応じて、システムの処理能力が最高になるように、プロセス毎のプロセッサモジュールを決め、プロセス間での処理を継続することにより、システム全体の処理能力を向上させる。

【０１４６】トランザクションプロセッサモジュール（１５）で障害が発生した場合を想定する。ここで、トランザクションプロセッサモジュール（１４～１６）とファイルプロセッサモジュール（１７～１９）はそれぞれ２台と３台で稼働すべきか、あるいは３台と２台で稼働すべきかを判定する。そして、システム処理能力の大きな方を選ぶ。

【０１４７】そして、前者の場合には、障害の発生したトランザクションプロセッサモジュールを閉塞する。後者の場合には、１つのファイルプロセッサモジュール（１９）がトランザクションプロセス（３１）を引き継ぐ。このように、各プロセッサモジュール（１１～１９）に複数のプロセス（３０～３２）を搭載させる構成も可能となる。

【０１４８】これにより、１つのトランザクションプロセス（３１）で障害が発生しても、プロセス単位のホットスタンバイ機能により、３つのトランザクションプロセス（３１）と２つのファイルプロセス（３２）といったシステムを柔軟に再構築することが可能となり、障害が発生しても、システム処理能力の低下を最小限にすることが可能となる。

【０１４９】

【発明の効果】本発明では、複数のプロセッサモジュールからなるシステムにおいて、受信側プロセッサモジュールが通信プロセッサモジュールにシステム稼働情報を送信し、通信プロセッサモジュールが処理時間の最小となる受信側プロセッサモジュールに電文を送信することにより、また、予備のプロセッサモジュールにも電文を送信することにより、プロセッサモジュールの負荷を均一にし、システム全体の処理能力が向上でき、更に、プロセッサモジュール障害時の回復時間を短縮できる。

【図面の簡単な説明】

【図１】本発明によるシステム構成図である。

【図２】本発明の処理概要を示す図である。

【図３】プロセッサモジュールの構成図である。

【図４】電文処理の概要を示す図である。

【図５】ソフトウェアの構成と電文処理の分割を示す図である。

【図６】IO制御装置のシステム構成図である。

【図７】プロセッサモジュールの状態遷移図である。

【図８】プロセッサモジュールの状態コードを示す図である。

【図９】制御用通信パスにおける電文フォーマットを示す図である。

【図１０】制御用電文種別を示す図である。

【図１１】システム稼働情報１とシステム稼働情報２を示す図である。

【図１２】電文のタイプとその個数を示す図である。

【図１３】電文のタイプとそのコードを示す図である。

【図１４】プロセッサモジュールとコンソールのアドレスを示す図である。

【図１５】プロセス間通信の電文フォーマットを示す図である。

【図１６】トランザクションプロセッサモジュールの入力待ち行列に格納されている電文を示す図である。

【図１７】ファイルプロセッサモジュールの入力待ち行列に格納されている電文を示す図である。

【図１８】トランザクションプロセッサモジュールの電文毎の処理ステップ数とファイルプロセッサモジュールとの通信回数を示す図である。

【図１９】ファイルプロセッサモジュールの電文毎の処理ステップ数とIO発行回数を示す図である。

【図２０】トランザクションプロセッサモジュールの処理能力と状態を示す図である。

【図２１】ファイルプロセッサモジュールの処理能力と状態を示す図である。

【図２２】システム稼働情報による処理手順を示す図である。

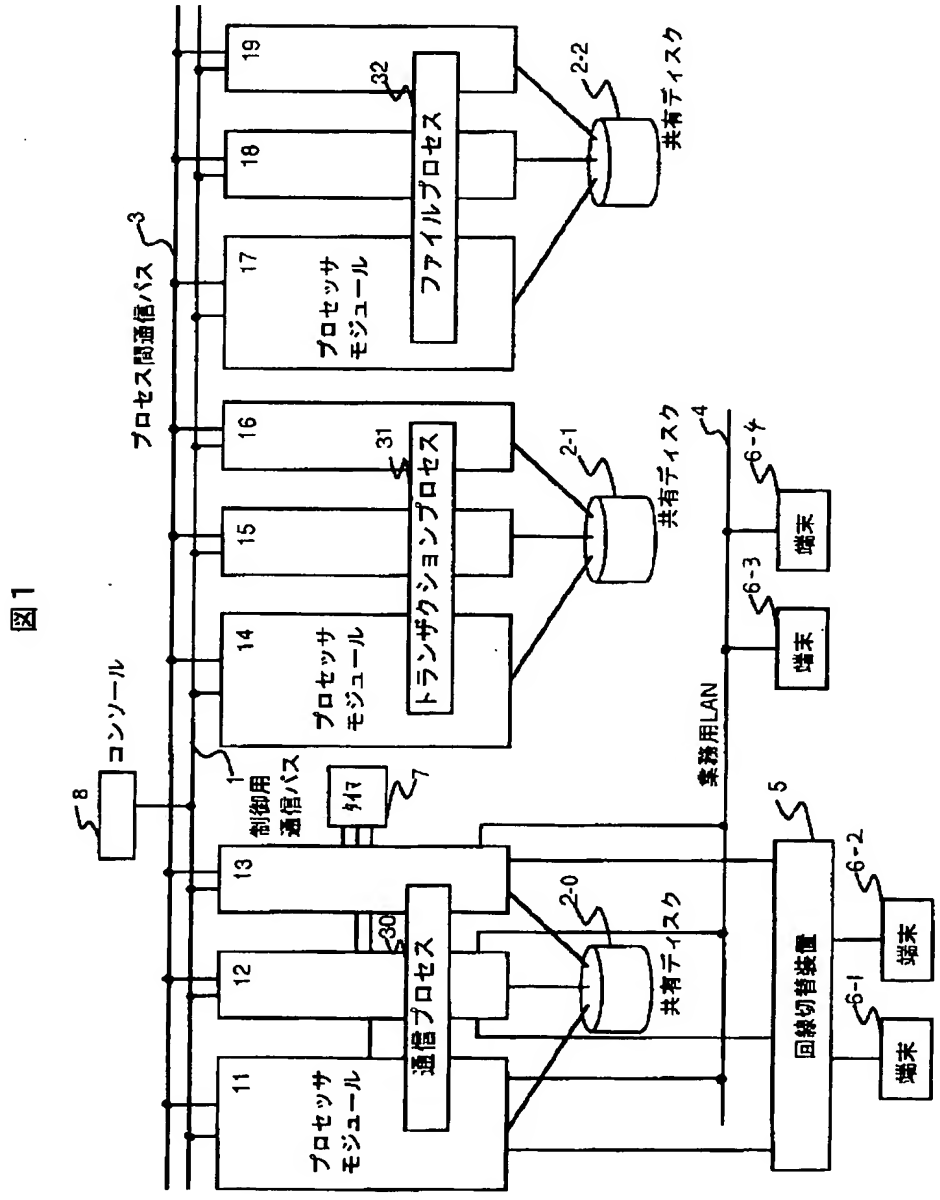
【図２３】通信プロセッサモジュールの処理時間の算出式である。

【図２４】制御用電文の一例を示す図である。

【図２５】障害時の処理手順を示す図である。

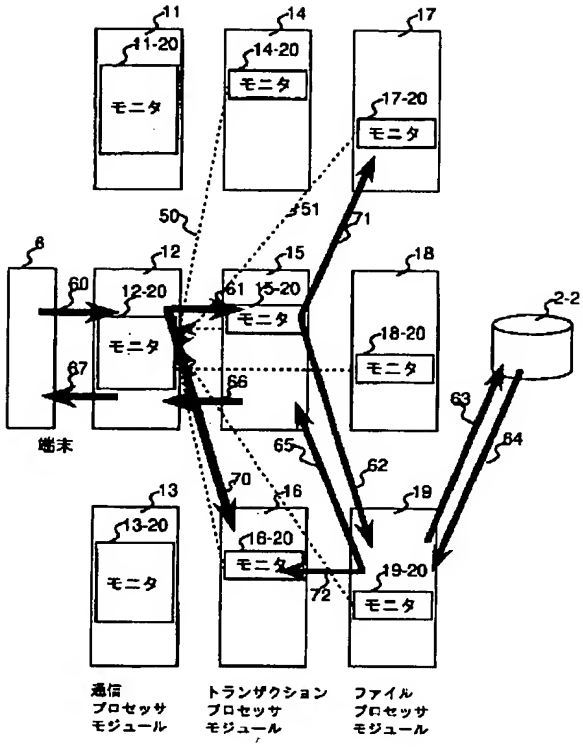
【図２６】障害回復時の処理手順を示す図である。
【図２７】障害時の制御用電文を示す図である。
【図２８】電文処理手順を示す図である。
【図２９】プロセス間通信用電文を示す図である。
【図３０】トランザクションプロセッサモジュールで障害が発生した場合の処理を示す図である。
【図３１】ファイルプロセッサモジュールで障害が発生した場合の処理を示す図である。
【図３２】プロセス間通信の電文を示す図である。
【図３３】予備のプロセッサモジュールを考慮した場合の処理を示す図である。
【図３４】予備のプロセッサモジュールを考慮した場合のプロセス間通信の電文を示す図である。
【図３５】トランザクションプロセッサモジュールで障害が発生した場合の処理を示す図である。
【図３６】ファイルプロセッサモジュールで障害が発生した場合の処理を示す図である。
【図３７】プロセス間通信の電文を示す図である。
【図３８】本発明によるシステム構成図（２）である。
【符号の説明】
１…制御用通信パス、２－０～２－２…共有ディスク、３…プロセス間通信パス、４…業務用LAN 、５…回線切替装置、６…端末、７…タイマ、１１～１９…プロセッサモジュール、３０…通信プロセス、３１…トランザクションプロセス、３２…ファイルプロセス

【図１】



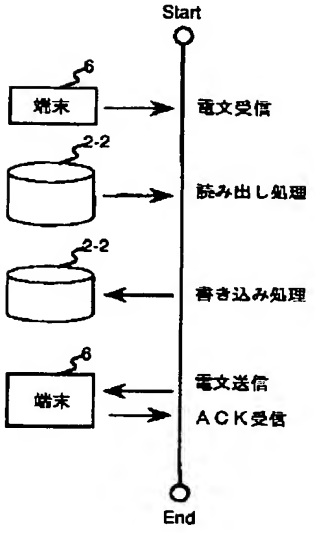
【図2】

図2



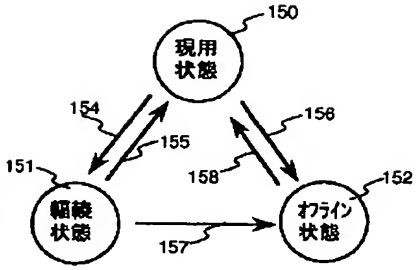
【図4】

図4



【図7】

図7



【図13】

図13

電文のタイプ	コード
電文1	(01)16
電文2	(02)16
電文3	(03)16
電文4	(04)16
電文5	(05)16

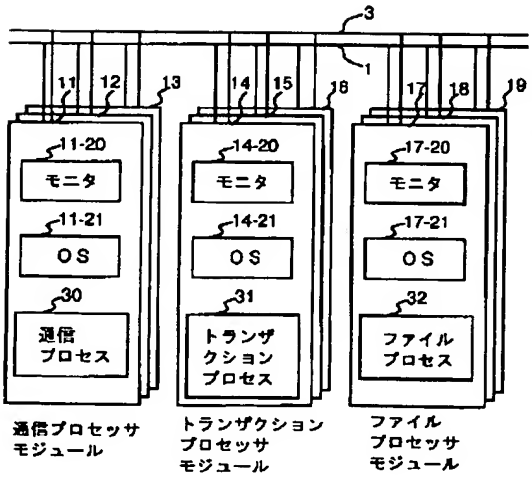
【図19】

図19

電文種別	実行ステップ数	通信回数	IO回数
	SF1(i)	CF1(i)	IF1(i)
電文1	100000	1	4
電文2	150000	1	5
電文3	130000	1	3
電文4	150000	1	4
電文5	170000	1	7

【図5】

図5



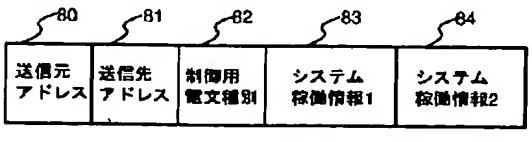
【図8】

図8

プロセッサモジュールの状態	コード
現用状態	(01)16
輻輳状態	(02)16
オフライン状態	(03)16

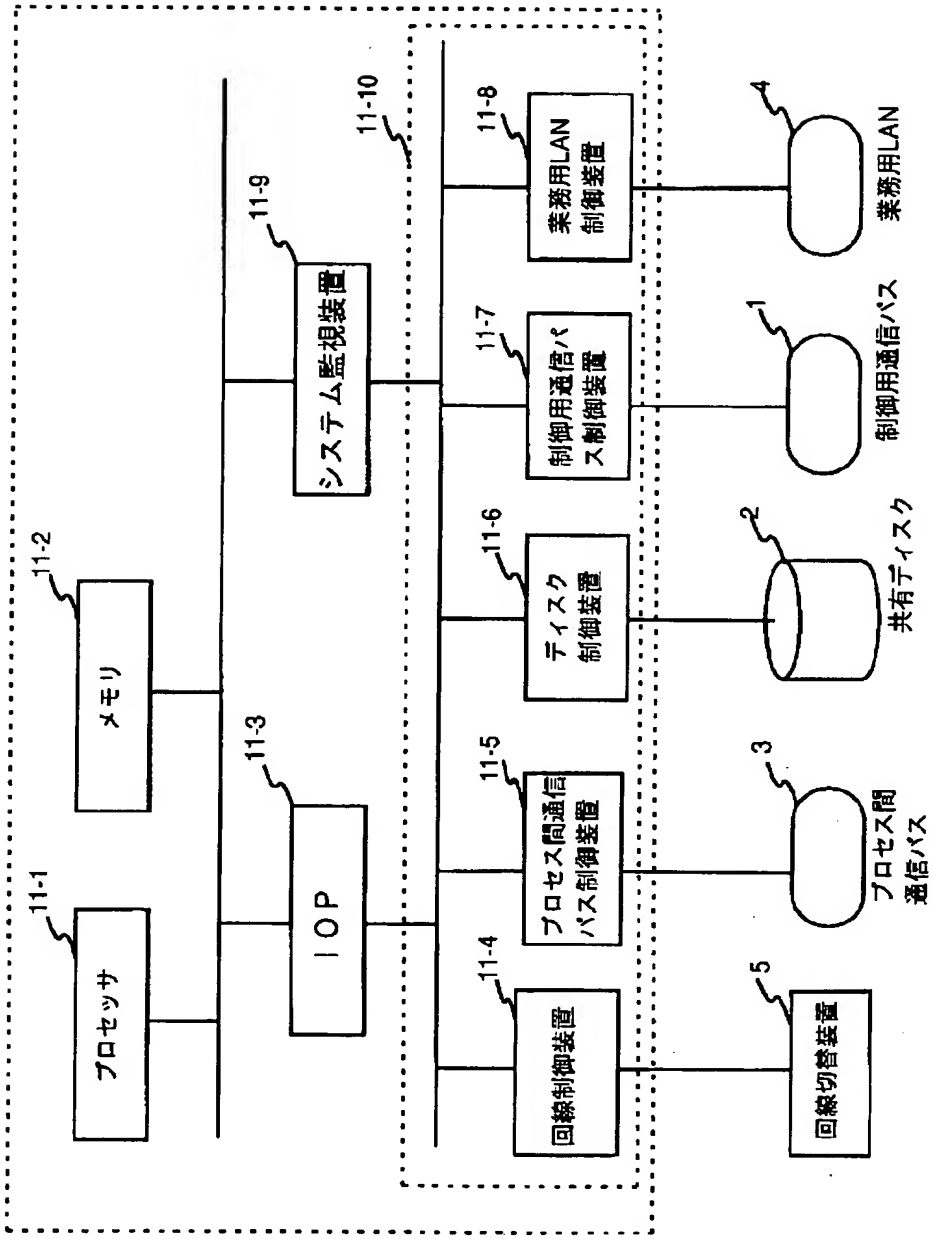
【図9】

図9



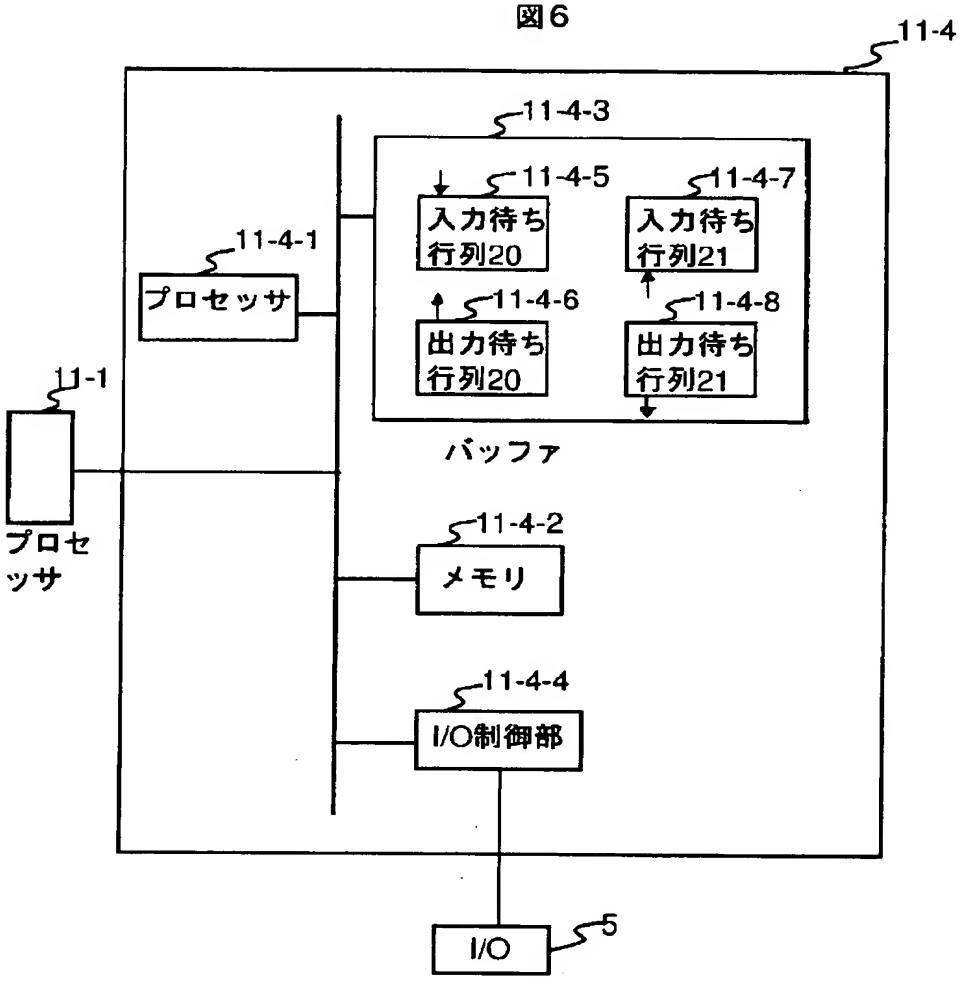
【図3】

図3


プロセッサ
11-1
メモリ
11-2
IOP
11-3
回線制御装置
11-4
プロセス間通信
パス制御装置
11-5
ディスク
制御装置
11-6
制御用通信パ
ス制御装置
11-7
システム監視装置
11-9
業務用LAN
制御装置
11-8
11-10
回線切替装置
5
プロセス間
通信パス
3
共有ディスク
2
制御用通信パス
1
業務用LAN
4

【図6】

図6



【図10】

図10

制御用電文	コード
リセットメッセージ	(10)16
aliveメッセージ	(01)16
システム稼働情報	(02)16

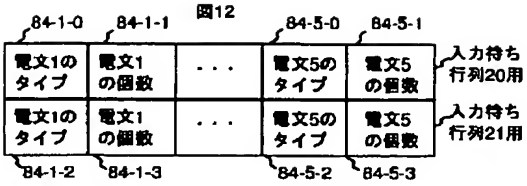
【図11】

図11

システム稼働情報1	システム稼働情報2	意　味
(01)16	図12参照	電文のタイプと個数
(10)16	(01)16	1つのプロセッサモジュールが輻輳発生
	(02)16	同一プロセスの全プロセッサモジュールで輻輳発生
	(03)16	障害発生の通知
	(10)16	輻輳解除通知
	(11)16	障害回復通知

【図12】

図12



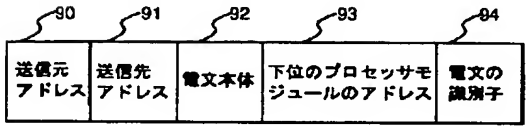
【図14】

図14

プロセッサモジュール	プロセッサモジュールのアドレス
プロセッサモジュール(11)	(00)16
プロセッサモジュール(12)	(01)16
プロセッサモジュール(13)	(02)16
プロセッサモジュール(14)	(10)16
プロセッサモジュール(15)	(11)16
プロセッサモジュール(16)	(12)16
プロセッサモジュール(17)	(20)16
プロセッサモジュール(18)	(21)16
プロセッサモジュール(19)	(22)16
コンソール	(30)16

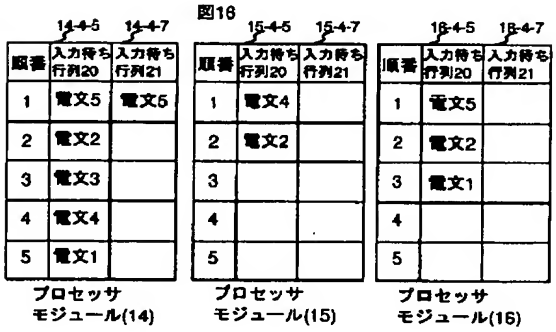
【図15】

図15



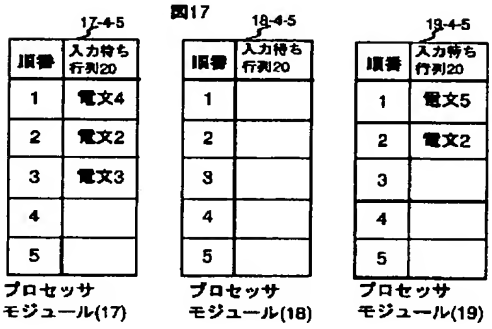
【図16】

図16



【図17】

図17



【図18】

図18

電文種別	実行ステップ数		通信回数	
	ST1(i)	ST2(i)	CT1(i)	CT2(i)
電文1	100000	100000	4	1
電文2	150000	150000	5	1
電文3	130000	130000	3	1
電文4	130000	130000	2	1
電文5	170000	170000	3	1

【図20】

図20

プロセッサモジュール	処理能力 PT(j)	状　態
プロセッサモジュール(14)	50000000	(02)16
プロセッサモジュール(15)	40000000	(01)16
プロセッサモジュール(16)	40000000	(01)16

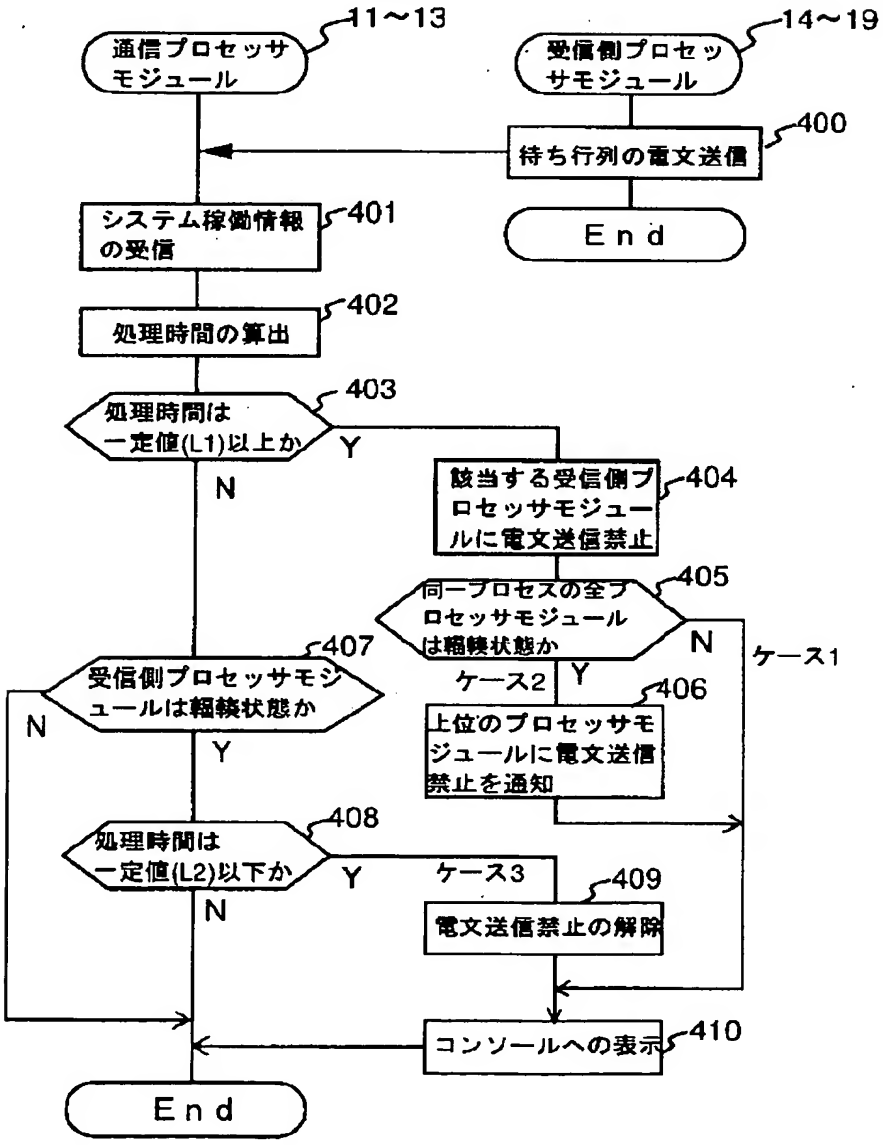
【図21】

図21

プロセッサモジュール	処理能力 PF(j)	状　態
プロセッサモジュール(17)	40000000	(01)16
プロセッサモジュール(18)	50000000	(03)16
プロセッサモジュール(19)	40000000	(01)16

【図22】

図22



【図23】

図23

(1)トランザクションプロセッサモジュール

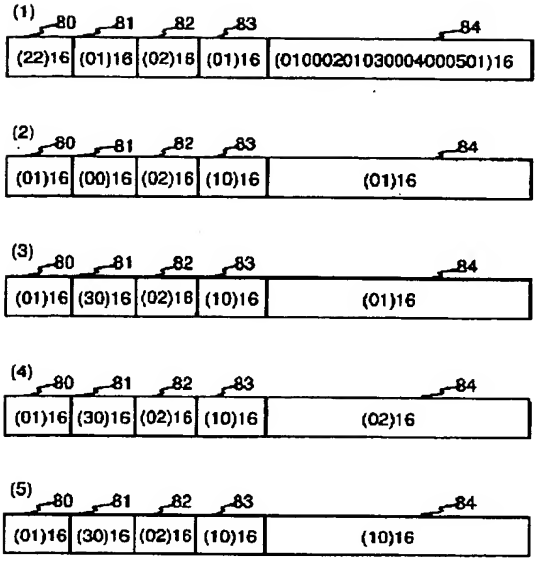
$$RT(j)=\frac{\sum_{i=1}^{5}\{ST1(i)\times NT1(i)+ST2(i)\times NT2(i)+\{CT1(i)+CT2(i)\}\times PS\}}{PT(j)}$$

(2)ファイルプロセッサモジュール

$$RF(j)=\sum_{i=1}^{5}\left(\frac{\{SF1(i)\times NF1(i)+CT1(i)\times PS\}}{PF(j)}+IF1(i)\times IT\right)$$

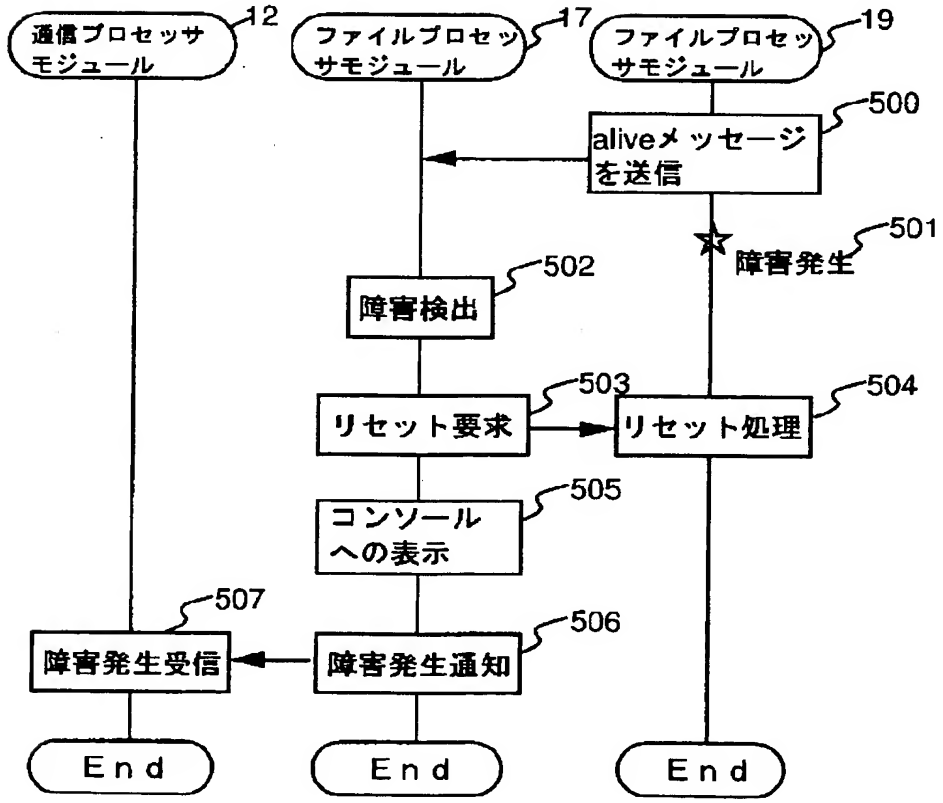
【図24】

図24



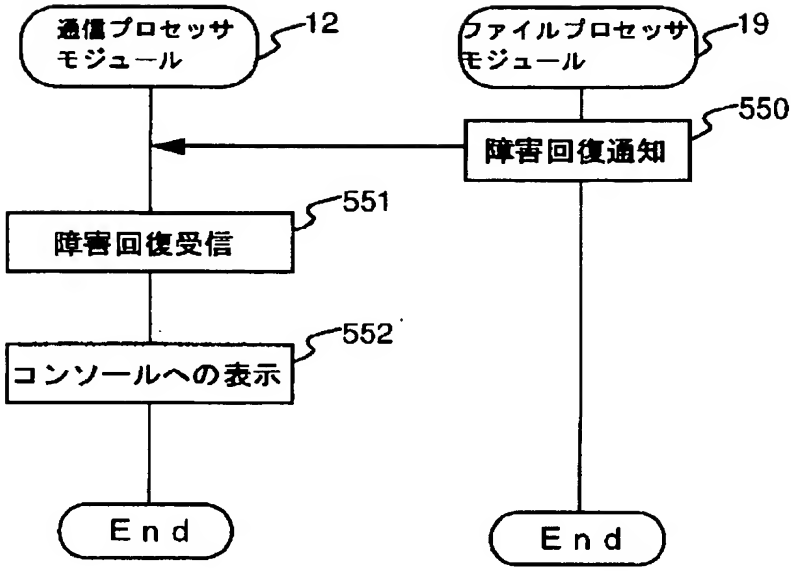
【図25】

図25



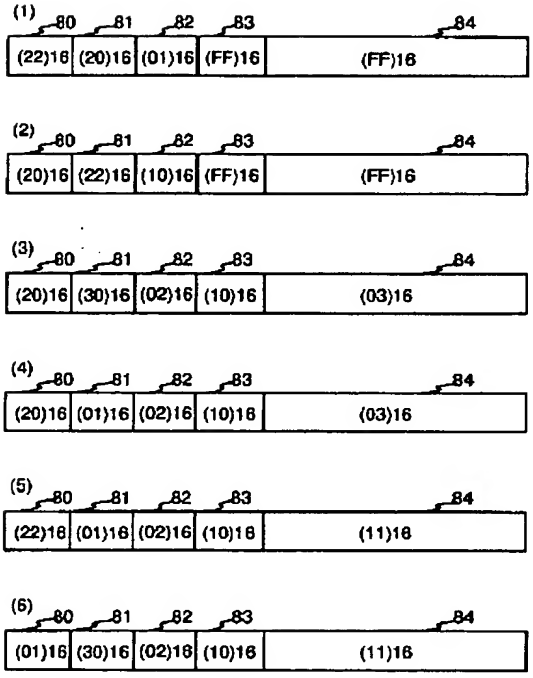
【図26】

図26



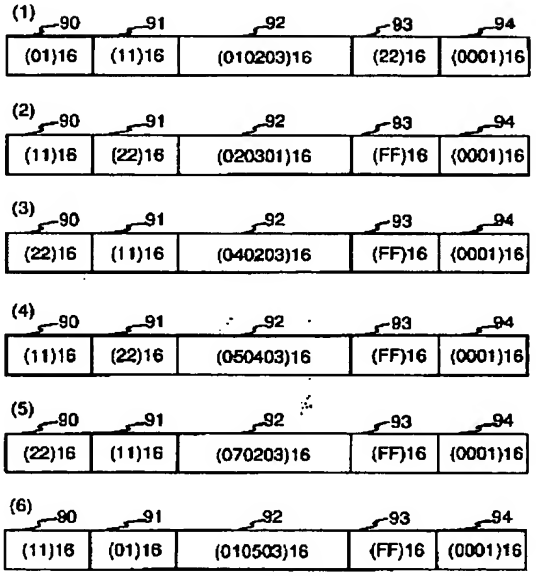
【図27】

図27



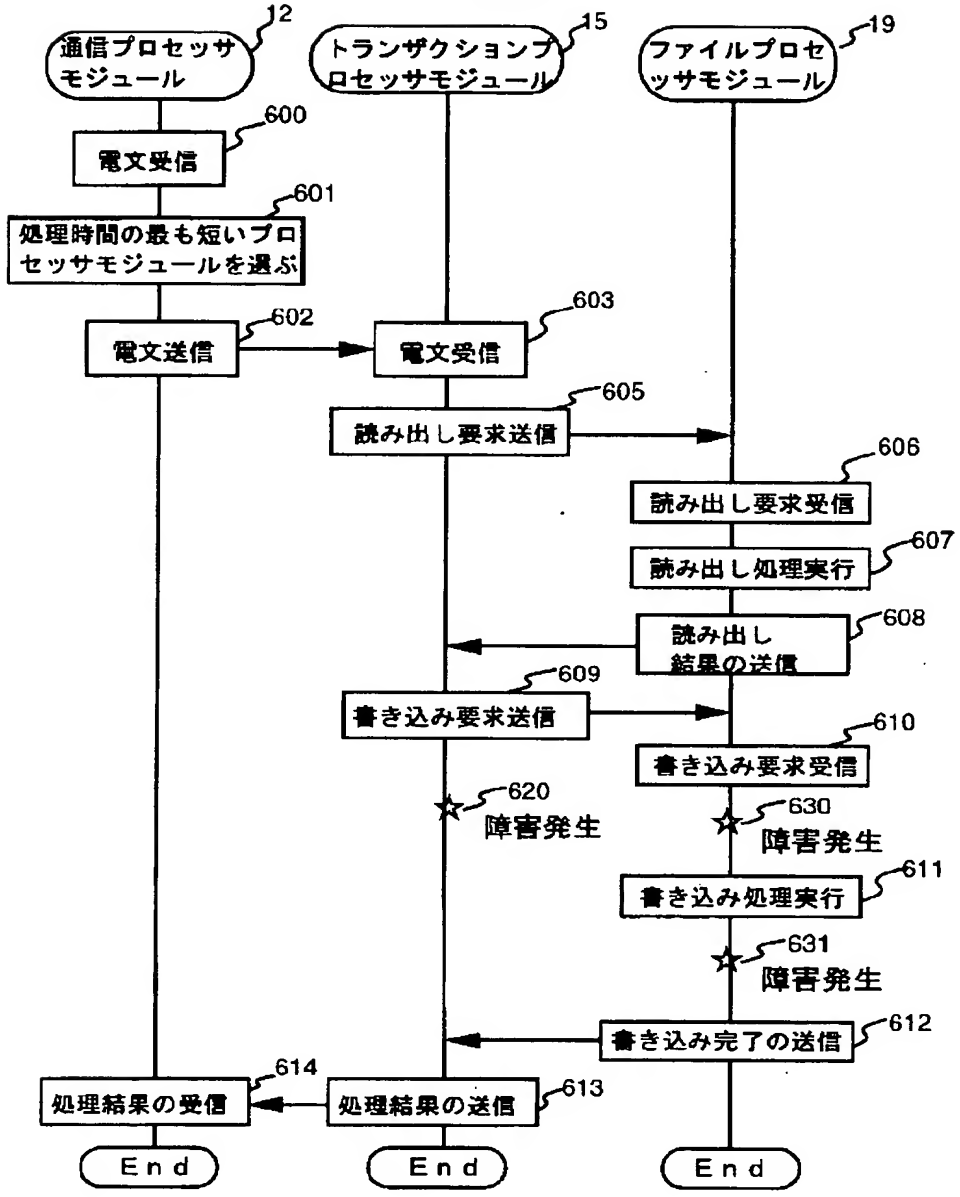
【図29】

図29



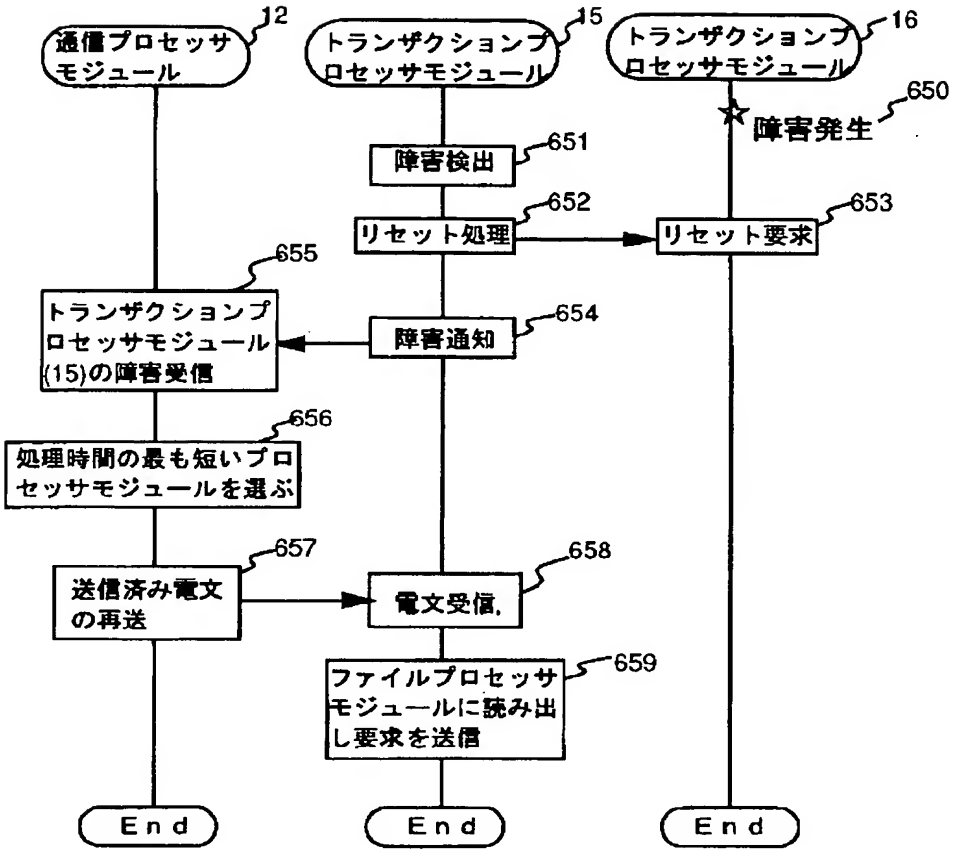
【図28】

図28


通信プロセッサモジュール
12
トランザクションプロセッサモジュール
15
ファイルプロセッサモジュール
19
電文受信
600
処理時間の最も短いプロセッサモジュールを選ぶ
601
電文送信
602
電文受信
603
読み出し要求送信
605
読み出し要求受信
606
読み出し処理実行
607
読み出し結果の送信
608
書き込み要求送信
609
書き込み要求受信
610
障害発生
620
障害発生
630
書き込み処理実行
611
障害発生
631
書き込み完了の送信
612
処理結果の受信
614
処理結果の送信
613
End
End
End

【図30】

図30



【図32】

図32

	80	81	82	83	84
(1)	(12)16	(11)16	(10)16	(FF)16	(FF)16
(2)	(12)16	(01)16	(02)16	(10)16	(03)16
(3)	(01)16	(11)16	(010203)16	(22)16	(0001)16
(4)	(20)16	(22)16	(10)16	(FF)16	(FF)16
(5)	(20)16	(01)16	(02)16	(10)16	(03)16
(6)	(01)16	(11)16	(010203)16	(20)16	(0001)16

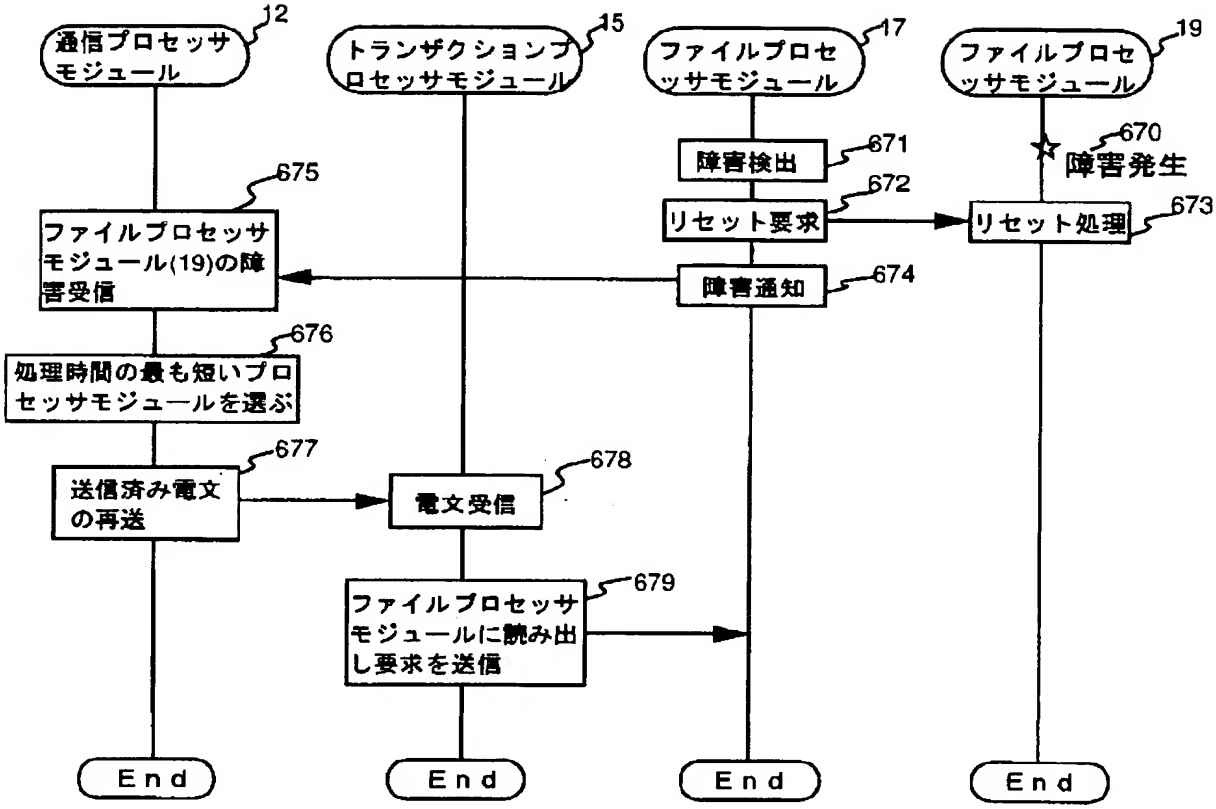
【図34】

図34

	90	91	92	93	94
(1)	(01)16	(1112)16	(010203)16	(2220)16	(0001)16
(2)	(11)16	(2220)16	(020301)16	(FF)16	(0001)16
(3)	(22)16	(1112)16	(040203)16	(FF)16	(0001)16
(4)	(11)16	(2220)16	(050403)16	(FF)16	(0001)16
(5)	(22)16	(1112)16	(070203)16	(FF)16	(0001)16
(6)	(11)16	(0102)16	(010503)16	(FF)16	(0001)16

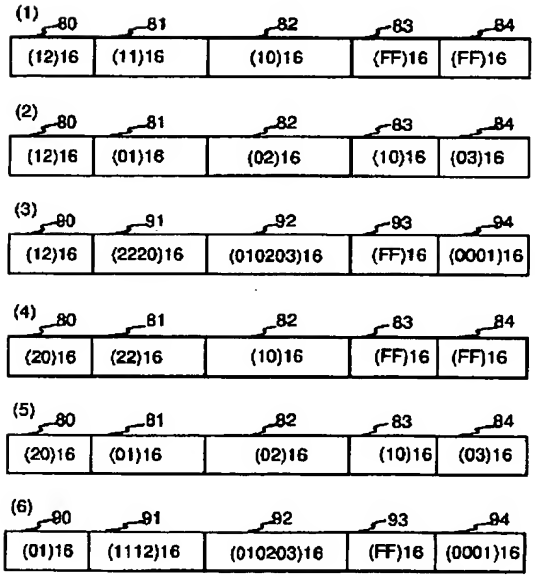
【図31】

図31



【図37】

図37



【図33】

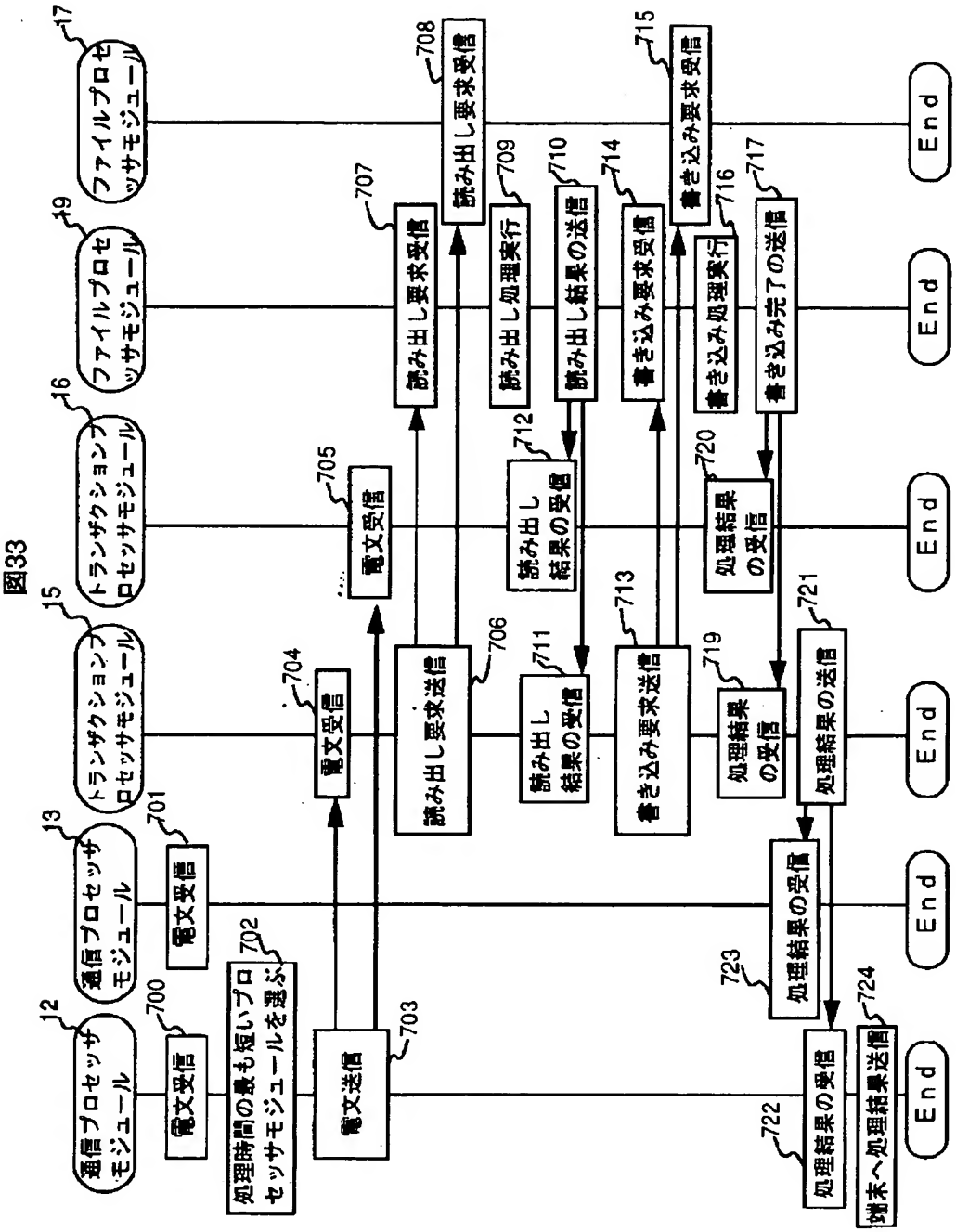


図33
12 通信プロセッサモジュール
13 通信プロセッサモジュール
15 トランザクションプロセッサモジュール
16 トランザクションプロセッサモジュール
19 ファイルプロセッサモジュール
17 ファイルプロセッサモジュール
700 電文受信
701 電文受信
702 処理時間の最も短いプロセッサモジュールを選ぶ
703 電文送信
704 電文受信
705 電文受信
706 読み出し要求送信
707 読み出し要求受信
708 読み出し要求受信
709 読み出し処理実行
710 読み出し結果の送信
711 読み出し結果の受信
712 読み出し結果の受信
713 書き込み要求送信
714 書き込み要求受信
715 書き込み要求受信
716 書き込み処理実行
717 書き込み完了の送信
719 処理結果の受信
720 処理結果の受信
721 処理結果の送信
722 処理結果の受信
723 処理結果の受信
724 端末へ処理結果送信
End

【図35】

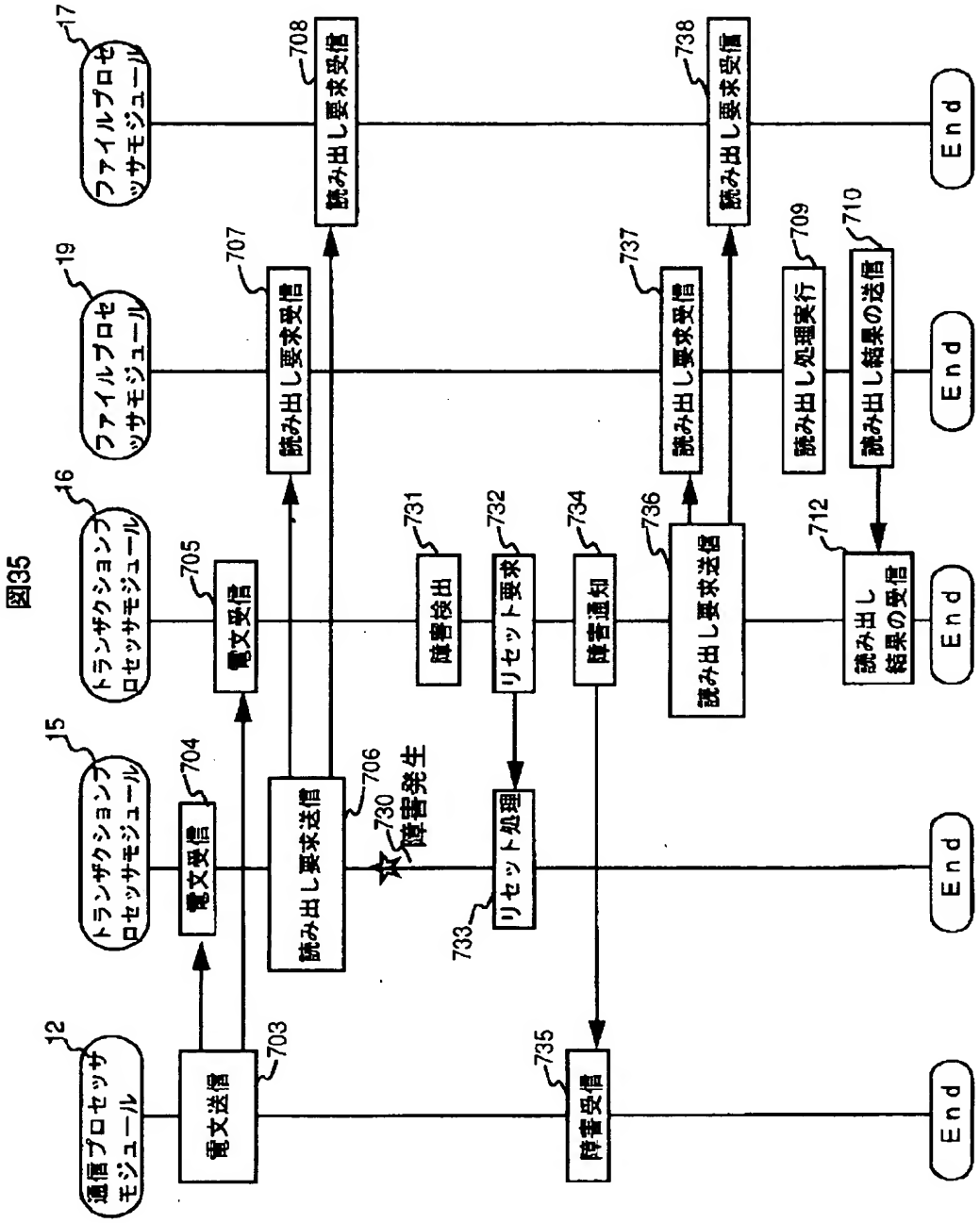


図35
通信プロセッサモジュール
12
トランザクションプロセッサモジュール
15
トランザクションプロセッサモジュール
16
ファイルプロセッサモジュール
19
ファイルプロセッサモジュール
17
電文送信
703
電文受信
704
電文受信
705
読み出し要求送信
706
読み出し要求受信
707
読み出し要求受信
708
障害発生
730
障害検出
731
リセット要求
732
リセット処理
733
障害通知
734
障害受信
735
読み出し要求送信
736
読み出し要求受信
737
読み出し要求受信
738
読み出し処理実行
709
読み出し結果の送信
710
読み出し結果の受信
712
End
End
End
End
End
End
End

【図36】

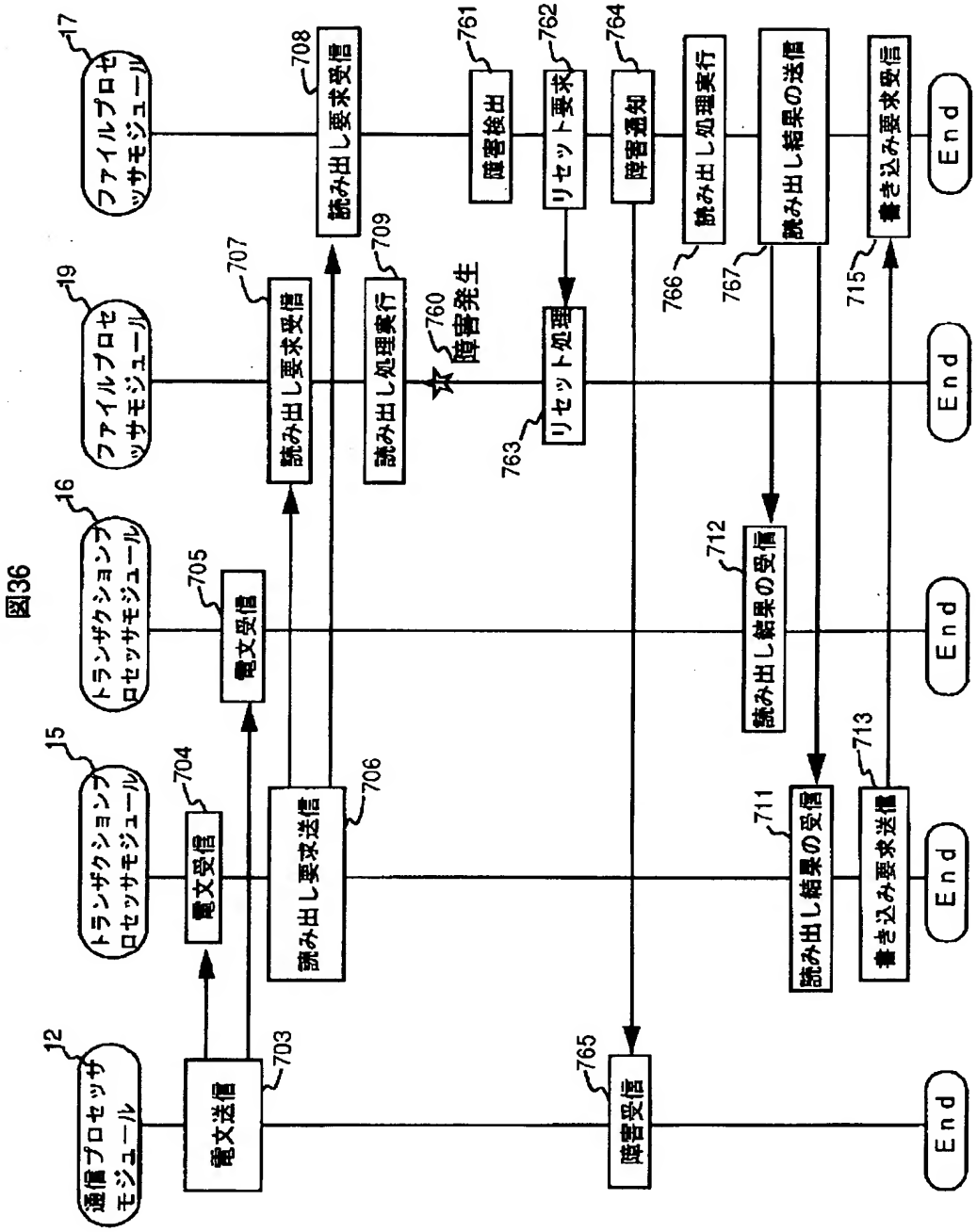
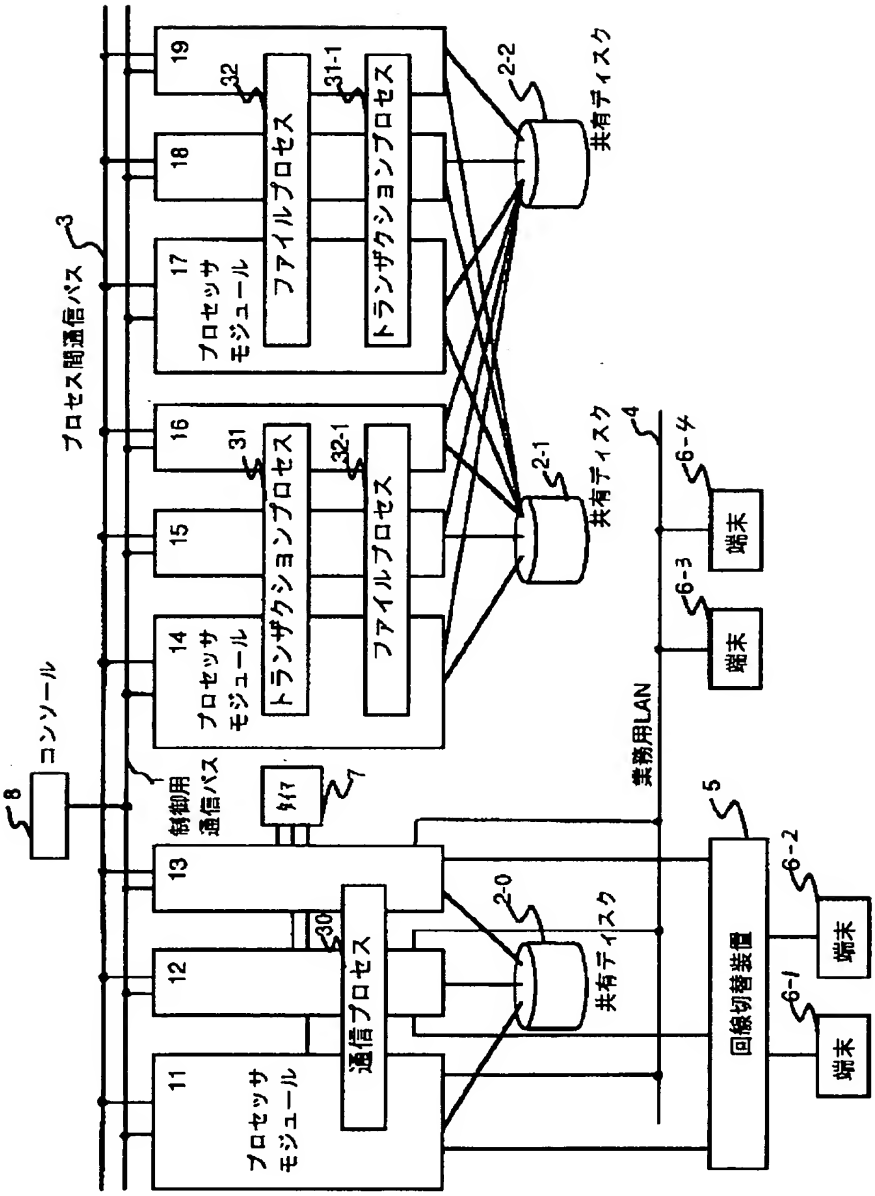


図36
通信プロセッサモジュール
12
ファイルプロセッサモジュール
17
ファイルプロセッサモジュール
19
トランザクションプロセッサモジュール
16
トランザクションプロセッサモジュール
15
電文送信
703
電文受信
704
電文受信
705
読み出し要求送信
706
読み出し要求受信
707
読み出し要求受信
708
読み出し処理実行
709
障害発生
760
障害検出
761
リセット要求
762
リセット処理
763
障害通知
764
障害受信
765
読み出し処理実行
766
読み出し結果の送信
767
読み出し結果の受信
711
読み出し結果の受信
712
書き込み要求送信
713
書き込み要求受信
715
End
End
End
End
End
End

【図38】

図38


プロセス間通信バス
3
ユンソール
8
制御用通信バス
7
業務用LAN
4
5
プロセッサモジュール
11
12
13
14
15
16
17
18
19
通信プロセス
30
トランザクションプロセス
31
ファイルプロセス
32
トランザクションプロセス
31-1
ファイルプロセス
32-1
共有ディスク
2-0
共有ディスク
2-1
共有ディスク
2-2
回線切替装置
端末
6-1
端末
6-2
端末
6-3
端末
6-4